AFRL-SN-WP-TR-2005-1113

INTEGRATED SENSING AND PROCESSING (ISP)

A Mathematical Methodology for Managing and Integrating Sensors and Processors in Distributed Systems for Radar and Communication

Chad M. Spooner

ATK Mission Research Corporation
10 Ragsdale Drive
Suite 210
Monterey, CA 93940

APRIL 2005

Final Report for 29 August 2002 – 28 February 2005

Approved for public release; distribution is unlimited

STINFO FINAL REPORT

SENSORS DIRECTORATE
AIR FORCE RESEARCH LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7320

NOTICE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the Air Force Research Laboratory Wright Site (AFRL/WS) Public Affairs Office (PAO) and is releasable to the National Technical Information Service (NTIS). It will be available to the general public, including foreign nationals.

PAO Case Number: AFRL/WS-05-2687, 29 Nov 2005

THIS TECHNICAL REPORT IS APPROVED FOR PUBLICATION.

/S/
Alan D. Kerrick, Project Engineer
RF Systems and Analysis Branch
RF Technology Division

/S/
Keith Loree, Branch Chief
RF Systems and Analysis Branch
RF Technology Division

/S/
Timothy R. Poth, Major, USAF
Deputy/Division Chief
RF Sensor Technology Division

This report is published in the interest of scientific and technical information exchange and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
April 2005	Final	29 Aug 2002 – 28 Feb 2005

4. TITLE AND SUBTITLE

Integrated Sensing and Processing (ISP)

A Mathematical Methodology for Managing and Integrating Sensors and Processors in Distributed Systems for Radar And Communications

5a. CONTRACT NUMBER
F33615-02-C-1198

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
69199F

6. AUTHOR(S)

Chad M. Spooner

5d. PROJECT NUMBER
ARPS

5e. TASK NUMBER
NR

5f. WORK UNIT NUMBER
0T

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

ATK Mission Research Corporation
10 Ragsdale Drive
Suite 210
Monterey, CA 93940

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

SENSORS DIRECTORATE

AIR FORCE RESEARCH LABORATORY

AIR FORCE MATERIEL COMMAND

WRIGHT-PATTERSON AFB, OH 45433-7320

10. SPONSOR/MONITOR'S ACRONYM(S)

AFRL/SNRR

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

AFRL-SN-WP-TR-2005-1113

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES

This document has color content.

14. ABSTRACT

The objective of this effort is to develop tools for integrating sensing and processing over as wide a range of application areas as possible. The approach is to consider systems of targets and sensors in as general a general mathematical formulation as possible, to develop mathematical tools to study such systems, and to apply the tools to problems in radar and communications. Accomplishments include results on the characterization of sources and sensors and decision-directed sensing and processing implemented through partially observed Markov decision processes (POMDPs) and binary hypertrees (BHCs). The characterization of sensors and sources shows that time-frequency distributions and wireless scattering functions can both be estimated by as a convolution of a Rihaczek time-frequency density with a time-frequency kernel function. POMDPs have been applied to a sensor-scheduling algorithm, and results indicate that the use of POMDPs may lead to much more efficient sensor network management. BHCs have been applied for the efficient solution of classifier problems.

15. SUBJECT TERMS:
integrated sensing and processing, sensor resource management, partially observed Markov decision processes, binary hypertrees

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Alan D. Kerrick
Unclassified	Unclassified	Unclassified	SAR	277	19b. TELEPHONE NUMBER *(include area code)* 937-255-6427 x 4343

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. 239.18

A Mathematical Methodology for Managing and Integrating Sensors and Processors in Distributed Systems for Radar and Communications (F33615–02–C–1198)

FINAL REPORT

Chad M. Spooner
ATK Mission Research

July 21, 2005

Abstract

The research and development objectives and results obtained over the course of MRC's *Integrated Sensing and Processing* contract are compiled in this document. The effort comprised several technical areas in radar- and communication-signal processing that could substantially benefit from an integration of sensors with processing. In particular, we studied problems related to the statistical and algebraic characterization of sources and sensors, the exploitation of channel and source statistics for improved communication in harsh environments, the determination of the minimum required link bandwidth and associated quantization scheme for transmission of local sensor estimates to other sensors or to a central processor, and problems in the area of decision-directed sensing and sensor-network management. In this latter technical area, we focused on the sensor-scheduling problem using partially observable Markov decision processes and on the problem of combining multiple single-modality target classifiers to form a hyperclassifier whose performance exceeds that of any of the constituent classifiers while minimizing the amount of data requested from the array of available sensor modalities.

Contents

1 Introduction

This document provides the final report for DARPA/AFRL contract F33615–02–C–1198 "Integrated Sensing and Processing (ISP)." The structure of the report is as follows. An executive summary is provided in Section 2, and Section 3 provides links to all the technical reports and published papers produced under the contract. If this report is being read electronically, these reports can also be viewed by clicking the colored text representing the report titles. Concluding remarks are made in Section 4.

2 Executive Summary

2.1 Overview of Project Objectives

The original objectives of the MRC-CSU ISP program are stated in the following list, which is taken from the technical proposal submitted to DARPA.

1. **Develop constituent mathematical tools for ISP.** Here the objective is to develop tools for integrating sensing and processing over as wide a range of application areas as possible. In particular, tools were to be developed relating to the following fundamental signal processing areas:

   (a) Characterization of sources and sensors.
   (b) Exploitation of channel and target statistics.
   (c) Sensor deployment and clustering.
   (d) Bandwidth allocation and information quantization.
   (e) Decision-directed sensing and processing.

2. **Construct a general ISP framework.** Here the idea is to refine the developed tools and create a framework for developing the interfaces between the tools, which can then be adapted to a specific problem of interest.

   (a) Refinement of developed mathematical tools.
   (b) Integration of mathematical tools into a general framework.
   (c) Validate framework.

3. **Apply ISP tools and framework to two problems in radar and communications.** In this third major objective, the idea is to apply the developed ISP tools and framework to each of two major problems of interest to the government: automatic target recognition and high-speed MIMO communication. The outcome would be a quantitative characterization of the performance or cost benefits of utilizing the new ISP tools in these familiar settings.

   (a) Radar-based automatic target recognition.
   (b) Multi-input multi-output communication link.

After the October 2003 ISP Program Review, DARPA decided not to exercise the FY04 and FY05 funding options for this effort. This lead to a serious descoping from our original objectives; the adjusted program scope is described in the following subsection. The reduced set of objectives was approved by the AFRL and DARPA program managers.

2.1.1 Program Objectives After Funding Decrease

1. **Develop constituent mathematical tools for ISP.**
   (a) Characterization of sources and sensors.
      i. Beamforming versus diversity combining, connections between radar scattering functions and time-frequency distribution analysis, connection between sensor-network control and radar-parameter adaptation [CSU].
   (b) Decision-directed sensing and processing.
      i. Partially observable Markov decision processes (POMDPs) for control and management of sensor networks [CSU].
      ii. Binary hypertrees for automatic target recognition [MRC].

2.2 Technical Approach

The technical approach employed for each of the three technical objectives is described at a high level in this section. The approaches employed for two additional objectives that were pursued prior to the descoping are also described. For more detail, please consult the appropriate technical report in Section 3.

2.2.1 Exploitation of Channel Statistics (Pre-Descoping)

We focus on the exploitation of communication-channel statistics to enable radio communication that is more robust to time-varying channel conditions, such as multipath and cochannel interference. The operational idea is to extend the notion of *rate-adaptive* communication links to *modulation-adaptive* communication links. In these latter links, many more system parameters are allowed to vary over time with respect to the former links, in which only the constellation of the employed digital QAM signal is allowed to vary. To develop this idea, we followed an information-theoretic technical approach. In particular, we developed an abstracted version of the problem and modeled the communication system and the physical channel as first-order Markov random processes. This leads to a model of the communication system that is constant over fixed periods of time, during which it is characterized as a discrete memoryless channel. We developed formulas for the capacity of such an adaptive system, which build on known formulas for a fixed system in the face of a time-varying physical channel.

2.2.2 Bandwidth Allocation and Quantization (Pre-Descoping)

The technical approach here is to formulate an optimization problem in which the sensors in a network can quantize their raw information and transmit it, or they can transform the information, then quantize and transmit. The idea is to determine whether any particular ordering of operations

is required for best performance. The results of this research can then be used in the design of quantization and transmission elements for wireless sensor networks. In particular, it may be quite beneficial for each sensor to quantize and transmit to a more sophisticated central processor rather than outfit the sensors themselves with sophisticated estimators and quantizers.

2.2.3 Characterization of Sources and Sensors

The technical approach for characterization of sources and sensors is to attempt to unify aspects of radar signal processing with wireless communication signal processing. In particular, an attempt is made to unify the ideas of time-frequency distributions and radar scattering functions. The benefit of this approach is that if a connection is forged, then the large body of work on time-frequency distributions may be brought to bear on the scattering-function estimation (channel estimation) problem.

2.2.4 POMDPs for Sensor Network Control and Management

The technical approach employed for sensor network control and management is first to focus on the *sensor scheduling* subproblem, and then to apply partially observable Markov decision processes to this subproblem. The scheduling problem is a serious one, involving as it does the ultimate tracking performance of the network as well as the lifespan and long-term utility of the network. Careful scheduling of the on and off states of the sensors can substantially increase the network lifetime and permit the network to be maximally useful (though degraded, perhaps) throughout its lifespan. The POMDP formulation of the problem uses particle filtering to estimate prior probabilities of the system state instead of assuming any particular probabilistic model, and this makes the approach much more realistic. The performance of the scheduler is compared to the closest-point-of-approach algorithm, a simple and popular alternative, which cannot make use of crucial sensor attributes such as individual error statistics, current power level remaining, cost of use, etc.

2.2.5 Binary Hypertree Classifiers for ATR

The technical approach here is to build on previously developed tree-based classifiers for ATR . This classifier approach employs the local discriminant basis (LDB) to automatically determine the best wavelet representation of the set of class inputs *for the purpose of classification*. This should be contrasted with the standard wavelet approach of finding the best wavelet representation (basis) for a set of inputs *for the purpose of compression.* To integrate sensing and processing, we envision an ATR system that has at its disposal several sensing modalities (e.g., different camera types or a set of distinct radar waveforms and bandwidths). A tree-based recognizer is constructed for each of the modalities and as many measurement functions (e.g., wavelet types) as desired, creating a family of tree-based classifiers. The hypertree idea is to link these trees together such that if an ambiguous node is reached in one tree, the node points to the tree having the best chance of removing the ambiguity. This tree is “jumped to” and if a new sensing is required, the data is obtained. In this way, the classification is performed in a sequential manner, and the best data subspace for classification is automatically determined by adaptively responding to the data.

2.3 Programmatic Approach

The programmatic approach involved the use of technical resources at MRC, Colorado State University (CSU), and a consultant. Funding was split nearly evenly between MRC and CSU. MRC was the prime contractor and CSU was the single subcontractor used for ISP. The single consultant was John Gubner.

The AFRL was MRC's immediate customer. The AFRL provided program management for several of the DARPA ISP awardees.

2.4 Accomplishments

The accomplishments of the contract involved technical progress on the three objectives involved in the descoped effort as well as progress on two additional objectives prior to the descoping. These are briefly described here at a high level. The accomplishments are described in more detail in the technical reports of Section 3.

1. **Exploitation of channel and target statistics (pre-descoping): ISP for generic communication links.** The capacity formula for a general time-varying digital communication system facing a time-varying physical channel–both modeled as Markov processes—was obtained. The ultimate capacity of such a system requires that the formula be evaluated for an infinite number of channel uses. We implemented the general formula in MATLAB and found that evaluating it was costly for even ten channel uses when the parameters of the Markov processes and the particular digital communication system parameters (e.g., alphabet size) were realistic. Nevertheless, we were able to show that the capacity of an ISP-enabled adaptive-modulation system can be orders of magnitude larger than that for a static system facing a time-varying channel. A complete technical report was prepared and submitted. It can be found in Section 3.2.
2. **Bandwidth allocation and quantization (pre-descoping): canonical coordinates for transform coding.** Assuming the additive white noise model for quantization, we have proved that the correct coordinate systems for quantization are the systems of half and full canonical coordinates. Half canonical coordinates minimize the trace of the error covariance matrix, while full canonical coordinates minimize the determinant. Others have previously proved that canonical coordinates are optimum for rank reduction as well. Together with our results, this means that we can first choose a coordinate system and then decide how many bits to spend on the components. See Section 3.3.
3. **Characterization of sources and sensors: time-frequency distributions and scattering functions.** We have studied the estimation of time-frequency distributions (TFDs) and estimation of wireless scattering functions (SFs). We have shown that the most general quadratic estimator of each that is delay- and modulation-invariant may be written as a convolution of a Rihaczek TF density with a TF kernel function. The representation illustrates a fundamental difference in the design aspects of the two problems. In TFD estimation, the Rihaczek TF density is the raw Rihaczek function for the time series, and the kernel is designed to convolve in time and frequency to perform a smoothing role. In SF estimation, the kernel is the true SF and the Rihaczek TF density is that for a transmitter signal designed to deconvolve in time and frequency to perform an inversion role. In each case, the obtained Fourier

transform identities in a four-corners diagram allow for kernel or Rihaczek design in the transformed space of ambiguity functions. See Section 3.1.

4. **Decision-directed sensing and processing: POMDPs for sensor network control.** We have developed a sensor-scheduling algorithm based on POMDPs. Instead of relying on analytic expressions for belief states, we use a Monte Carlo approach that combines particle filtering for non-Gaussian nonlinear belief-state estimation with a Q-value approximation method that allows long-term look-ahead and thereby avoids producing greedy algorithms. Our algorithm is shown to outperform the closest-point-of-approach algorithm using simulated data. This indicates that the use of POMDPs may lead to much more efficient sensor network management, and therefore much longer-lived sensor networks. See Section 3.4.1.

5. **Decision-directed sensing and processing: binary hypertrees for ATR.** In the first part of this work, we established the mathematical foundations for representing and analyzing binary-tree classifiers that are based on exploitation of the LDB. We defined three basic classifier types:

   (a) The binary tree classifier (BTC). This classifier is associated with a single modality and a single measurement function (e.g. wavelet type).

   (b) The binary hypertree classifier (BHC). This classifier is comprised of a linked set of BTCs. As such, it encompasses multiple modalities and/or multiple measurement functions.

   (c) The binary supertree classifier (BSC). This classifier represents optimal (fusion) performance. It is a BTC but its input is the concatenation (or tiling for two-dimensional inputs) of all available modalities.

   The mathematical work established the basic performance ordering of $\{\text{BTC}\} \leq \text{BHC} \leq \text{BSC}$. The mathematics of this work were submitted in a technical report and can be found in Section 3.

   In the second part of the hypertree work, we implemented the three basic classifiers in MATLAB to provide a proof of concept and to evaluate performance claims made in the analytical work. We studied the performance of the classifiers using one- and two-dimensional synthetic problems and by applying them to several collected public data sets. A key accomplishment is the construction of an algorithm to automatically jointly determine a good (BTC) tree topology and classifier parameters for an arbitrary classification problem. We found that the BTC was able to embody the essential class ambiguity structure of the problems under study.

   For the one-dimensional problem, which involved the sixteen maximal-length shift-register (MLSR) sequences for shift-register length eight, we found that the BTCs, BHC, and BSC all delivered good-to-excellent performance and that performance was dependent on the particular wavelet type.

   For the two-dimensional problem, which involved four modalities and eight classes, we found that the predicted performance ordering held in all cases and that performance was not particularly sensitive to the wavelet type. This is due to the presence of severe class ambiguities (large equivalence classes) for each of the modalities. See Section 3.4.2.

3 Technical Reports and Publications

When viewing this document electronically, click on the title of the desired technical document to view it in your default PDF-file viewer. The report PDF files are contained in subdirectories of the directory containing this document. If you are reading a printed form of the document, each report is an appendix, and the page number for the appendix is provided next to the document title in the list below.

3.1 Algebraic Characterization of Sources and Sensors

1. "Estimating Time-Frequency Distributions and Scattering Functions Using the Rihaczek Distribution" [1]. See Appendix A, page 11.

2. "The Susman, Moyal, and Janssen Formulas Follow as Fourier Transform Identities . . . " [2]. See Appendix B, page 16.

3. "ISP Technical Report". See Appendix C, page 17.

3.2 Exploitation of the Statistics of Propagation Channels and Targets

1. "ISP for Communication Links: Mathematical Modeling, Problem Formulation," and Analysis" [4]. See Appendix D, page 40.

3.3 Bandwidth Allocation and Quantizing

1. "Canonical Coordinates for Transform Coding of Random Sources from Noisy Observations" [5]. See Appendix E, page 77.

2. "Canonical Coordinates are the Right Coordinate System for Transform Coding of Noisy Sources" [6]. See Appendix F, page 96.

3.4 Decision-Directed Sensing and Processing

3.4.1 Sensor Management and Control

1. "Sensor Scheduling for Target Tracking: A Monte Carlo Sampling Approach" [9]. This work was also published, with small differences, in [7] and [8]. See Appendix G, page 100.

3.4.2 Hypertree Classifiers

1. "Binary Hypertree Classifiers for ATR: Definitions, Analysis, and Algorithms" [10]. See Appendix H, page 114.

2. "Binary Hypertree Classifiers for ATR: Experimental Study" [11]. See Appendix I, page 160.

3.5 Applications of the Mathematical Methodology

Due to the funding constraints imposed on MRC during the performance of this contract, the developed algorithmic technology was not applied to any real-world defense-related collected data sets.

4 Conclusions

We make the following conclusions based on our ISP work under this contract.

1. For general communication links, a large increase in capacity can be obtained by employing the ISP-inspired notion of modulation adaptation. This notion generalizes rate-adaptive signaling to modulation-adaptive signaling by allowing multiple aspects of the transmitted waveform to be adapted, such as the transmission band, modulation type, coding, and modulation rates.
2. For sensor networks, partially observable Markov decision processes (POMDPs) appear to be very useful for the sensor-scheduling problem. In particular, this approach naturally allows various real-world sensor constraints, such as battery life, current power level, cost of operation, etc., to be taken into account in a dynamic fashion. Such an approach leads to more efficient use of the total network resources, and can thereby extend the useful lifespan of a sensor network.
3. For automatic target recognition, we have found that the notion of hypertree classification has significant merit. In particular, the developed method automatically finds the best sequential classifier that can be built from the collection of constituent classifiers, any one of which can be arbitrarily bad. A significant outcome of this work is the development of an automated training algorithm for jointly determining the tree topology, class splits, and feature vectors for an arbitrary classification problem.

References

[1] D.C. Farden and L.L. Scharf, “Estimating Time-Frequency Distributions and Channel Scattering Functions using the Rihaczek Distribution,” IEEE Sensor Array and Multichannel Signal Processing Workshop, Sitges, Spain, July 18, 2004.

[2] D.C. Farden and L.L. Scharf, “The Sussman, Moyal, and Janssen Formulas follow as Fourier Transform Identities of a More Fundamental Convolution Identity,” IEEE Signal Processing Magazine, submitted Sept 2004.

[3] J. Gubner, “Technical Report for Summer 2003, Agreement No: CSL-02394.01:JAG,” Consultant Technical Memorandum for DARPA ISP, October 2003.

[4] C. M. Spooner, “ISP for Communication Links: Mathematical Modeling, Problem Formulation, and Analysis,” MRC Technical Memorandum for DARPA ISP, May 2003.

[5] P.J. Schreier, L.L. Scharf, and T.J. Hu ”Canonical Coordinates for Transform Coding of Random Sources from Noisy Observations,” IEEE Trans Signal Processing, submitted, July 14, 2003, revised Nov 2004, to appear in 2005.

[6] P.J. Schreier, L.L. Scharf, T. Hu, S.D. Voran, ”Canonical coordinates are the right coordinate system for transform coding of noisy sources,” in Proc. IEEE Workshop on Statistical Signal Processing, pp. 221–224, St. Louis, MO, Sept 28-Oct 1,2003,

[7] Y. He and E. K. P. Chong, ”Sensor scheduling for target tracking in sensor networks,” in Proceedings of the 43rd IEEE Conference on Decision and Control (CDC’04), Atlantis Resort, Paradise Island, Bahamas, December 14–17, 2004, pp. 743–748.

[8] Y. He and E. K. P. Chong, ”Sensor scheduling for target tracking: A Monte Carlo sampling approach,” in Proceedings of the 2004 Workshop on Defense Applications of Signal Processing (DASP’04), The Homestead Resort, Midway, Utah, (originally scheduled for October 31–November 5, 2004; currently postponed) (Invited paper), to appear.

[9] Y. He and E. K. P. Chong, ”Sensor scheduling for target tracking: A Monte Carlo sampling approach,” submitted to Digital Signal Processing.

[10] C. M. Spooner, “Binary Hypertree Classifiers for ATR: Defintions, Analysis, and Algorithms,” MRC Technical Memorandum for DARPA ISP, March 2004.

[11] C. M. Spooner, “Binary Hypertree Classifiers for ATR: Experimental Study,” MRC Technical Memorandum for DARPA ISP, March 2005.

ESTIMATING TIME-FREQUENCY DISTRIBUTIONS AND SCATTERING FUNCTIONS USING THE RIHACZEK DISTRIBUTION

David C. Farden
ECE Department
North Dakota State University
Fargo, North Dakota 58105
david.farden@ndsu.nodak.edu

Louis L. Scharf
ECE and Statistics Departments
Colorado State University
Fort Collins, Colorado 80523-1373
scharf@engr.colostate.edu

ABSTRACT

In this paper we study two problems: estimation of time-frequency distributions (TFDs) and estimation of wireless or radar scattering functions (SFs). We show that the most general quadratic, delay- and modulation-invariant estimator of each may be written as a convolution of a Rihaczek TF density with a TF kernel. This representation complements other equivalent representations and establishes a fundamental connection between the *analysis* of the two problems. However, the representation also illustrates a fundamental difference in *design* for the two problems. For TFD estimation, the Rihaczek TF density is the raw Rihaczek for the time series and the kernel is designed to *convolve* in time and frequency for *smoothing*. For SF estimation, the kernel is the true SF and the Rihaczek TF density is that for a transmitter signal designed to *deconvolve* in time and frequency for *inversion*. In each case, the Fourier transform identities in a four-corners diagram allow for kernel or Rihaczek design in the transformed space of ambiguity functions. Design in this space then produces spectrograms and interferograms for TFD estimation, and ideal transmitter signals for SF estimation.

1. INTRODUCTION

In this paper we offer yet another representation for the Cohen class [1] of quadratic, delay- and modulation-invariant time-frequency distributions (TFDs), this one based on a convolution of the Rihaczek TF density with a TF kernel. This representation is used to produce spectrograms and interferograms which are practical estimators of TFDs.

We then ask whether this representation has any relevance to the estimation of scattering functions (SFs) for wireless and radar channels. The answer is yes. In fact, by following Gaarder's original arguments [2], we find that the most general quadratic, delay- and modulation-invariant estimator of the SF has the same representation, namely a convolution of a Rihaczek TF density with a TF kernel. But here the similarities end, for there is a key difference in *design* philosophy for TFD estimation and SF estimation.

In TFD estimation, the Rihaczek density is the raw Rihaczek for the time series, and the TF kernel is a free design variable. This kernel is designed to *smooth* in time and frequency. In SF estimation, the TF kernel is the true SF and the Rihaczek TF density is a free variable. This Rihaczek is actually the Rihaczek for the transmitted signal, which is designed to *invert* in time and frequency for the true SF. Thus, for TFD estimation the problem is one of design for *convolution*, whereas for SF estimation the problem is one of design for *deconvolution.* In each case, the Fourier transform identities in a four-corners diagram allow for kernel or Rihaczek design in the transformed space of ambiguity functions. Design in this space then produces spectrograms and interferograms for TFD estimation, and ideal transmitter signals for SF estimation.

The equations we derive are

$$\hat{P}_{xx}(t,f) = \int\int X(f')e^{j2\pi f't'}x^*(t')e(t-t',f-f')df'dt'$$

for TFD estimation, and

$$\hat{P}_{\sigma}(\tau,\nu) = \int\int X(\nu')e^{j2\pi\nu'\tau'}x^*(\tau')P_{\sigma}(\tau-\tau',\nu-\nu')d\nu'd\tau'$$

for SF estimation. Thus, for TFD estimation the problem is to design the kernel $e(t,f)$ so that the raw Rihaczek $X(f)e^{j2\pi ft}x^*(t)$ is smoothed, whereas for SF estimation the problem is to design the transmitted signal $x(t)$ so that its raw Rihaczek will invert for the unknown scattering function $P_{\sigma}(\tau,\nu)$ (also called the channel ambiguity function). Thus, while the *design* objectives are different, the defining *analysis* equations are identical. The transform duals of these equations are

$$\hat{\Gamma}_{xx}(\Delta f,\Delta t) = w(\Delta f,\Delta t)\Gamma_{xx}(\Delta f,\Delta t)$$

and

$$E(\Gamma_{yy}(\Delta f,\Delta t)) = R_H(\Delta f,\Delta t)\Gamma_{xx}(\Delta f,\Delta t),$$

where Γ_{xx}, w, and R_H are the 2D Fourier transforms of $X(f)e^{j2\pi ft}x^*(t)$, e, and P_{σ}, respectively. For TFD estimation, $w(\Delta f,\Delta t)$ is designed, and for SF estimation, $\Gamma_{xx}(\Delta f,\Delta t)$ is designed.

2. RIHACZEK FOUR-CORNERS DIAGRAM

A four-corners diagram can be used to illustrate relationships between time-frequency distributions, time-varying system representations, as well as related correlations and convolutions. Consider the four-corners diagram in Figure 1. We begin in the East with the Rihaczek [3] complex cross-energy density,

$$\gamma_{f,g}(t,f) = F(f)e^{j2\pi ft}g^*(t). \quad (1)$$

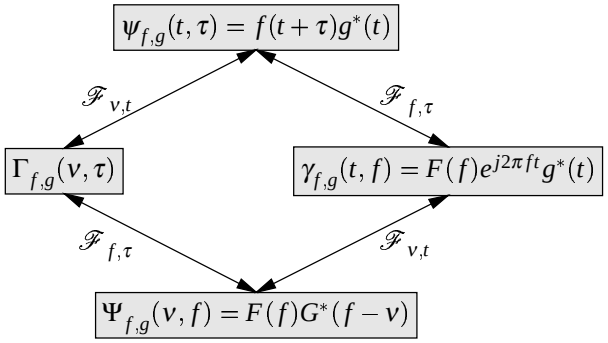


Figure 1: Rihaczek four-corners diagram.

With $\mathscr{F}^{-1}_{\tau,f}$ denoting the inverse Fourier transform operator from f to τ, we may move North to produce

$$\psi_{f,g}(t,\tau) = \mathscr{F}^{-1}_{\tau,f}\gamma_{f,g}(t,f) = f(t+\tau)g^*(t). \qquad (2)$$

Moving from East to South,

$$\Psi_{f,g}(\nu,f) = \mathscr{F}_{\nu,t}\gamma_{f,g}(t,f) = F(f)G^*(f-\nu), \qquad (3)$$

and from North to West to find the cross-ambiguity function for f and g,

$$\Gamma_{f,g}(\nu,\tau) = \mathscr{F}_{\nu,t}\psi_{f,g}(t,\tau) = \int_{-\infty}^{\infty} f(t+\tau)g^*(t)e^{-j2\pi\nu t}\,dt. \qquad (4)$$

The move from South to West yields an equivalent expression for the cross-ambiguity function:

$$\Gamma_{f,g}(\nu,\tau) = \mathscr{F}^{-1}_{\tau,f}\Psi_{f,g}(\nu,f) = \int_{-\infty}^{\infty} F(f)G^*(f-\nu)e^{j2\pi f\tau}\,df. \qquad (5)$$

Defining the modulation and time-shift operators M_{f_o} and T_{t_o} as $M_{f_o}g(t) = g(t)e^{j2\pi f_o t}$ and $T_{t_o}g(t) = g(t-t_o)$, we easily find covariant connections

$$\Gamma_{M_{f_o}f,g}(\nu,\tau) = e^{j2\pi f_o\tau}\Gamma_{f,g}(\nu-f_o,\tau),$$
$$\Gamma_{T_{t_o}f,g}(\nu,\tau) = \Gamma_{f,g}(\nu,\tau-t_o),$$
$$\Gamma_{f,M_{f_o}g}(\nu,\tau) = \Gamma_{f,g}(\nu+f_o,\tau),$$
$$\Gamma_{f,T_{t_o}g}(\nu,\tau) = e^{-j2\pi\nu t_o}\Gamma_{f,g}(\nu,\tau+t_o),$$
$$\Gamma_{M_{f_a}f,M_{f_b}g}(\nu,\tau) = e^{j2\pi f_a\tau}\Gamma_{f,g}(\nu+f_b-f_a,\tau),$$

and $\Gamma_{T_{t_a}f,T_{t_b}g}(\nu,\tau) = e^{-j2\pi\nu t_b}\Gamma_{f,g}(\nu,\tau+t_b-t_a)$.

The Fourier Transform preserves inner products. Since $\Gamma_{f,g}(\nu,\tau) = \mathscr{F}_{\nu,t}\mathscr{F}^{-1}_{\tau,f}\gamma_{f,g}(t,f)$, we find $\langle\gamma_{f,g},\gamma_{y,x}\rangle = \langle\Gamma_{f,g},\Gamma_{y,x}\rangle$, where

$$\langle\gamma_{f,g},\gamma_{y,x}\rangle = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(\gamma_{f,g}\gamma^*_{y,x})(t,f)\,dt\,df = \langle f,y\rangle\langle g,x\rangle^*. \qquad (6)$$

This result is known as Moyal's formula [4]:

$$\langle\Gamma_{f,g},\Gamma_{y,x}\rangle = \langle\gamma_{f,g},\gamma_{y,x}\rangle = \langle f,y\rangle\langle g,x\rangle^*. \qquad (7)$$

3. THE SUSSMAN FOUR-CORNERS DIAGRAM

Sussman [5] was apparently the first to state a particularly useful identity for ambiguity functions. Consider the Sussman four-corners diagram in Figure 2.

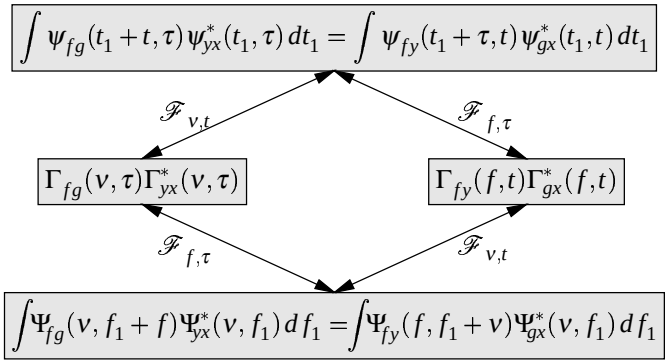


Figure 2: Sussman four-corners diagram.

Begin with the conjugate product in the West

$$w(\nu,\tau) = \Gamma_{f,g}(\nu,\tau)\Gamma^*_{y,x}(\nu,\tau). \qquad (8)$$

Since the North function is $n(t,\tau) = \mathscr{F}^{-1}_{t,\nu}w(\nu,\tau)$, we know that $n(t,\tau)$ is the following correlation in t and conjugate product in τ (refer also to Figure 1):

$$\begin{aligned} n(t,\tau) &= \int_{-\infty}^{\infty}\psi_{f,g}(t'+t,\tau)\psi^*_{y,x}(t',\tau)\,dt' \\ &= \int_{-\infty}^{\infty} f(t'+t+\tau)g^*(t'+t)y^*(t'+\tau)x(t')\,dt' \\ &= \int_{-\infty}^{\infty}\psi_{f,y}(t'+\tau,t)\psi^*_{g,x}(t',t)\,dt'. \end{aligned}$$

Note that to obtain the last line from the first above, exchange t with τ and g with y. Similarly, the remaining corners of Figure 2 can be obtained. The relationship between the West and the East of Figure 2 is known as the Sussman identity:

$$\mathscr{F}_{f,\tau}\mathscr{F}^{-1}_{t,\nu}\Gamma_{f,g}(\nu,\tau)\Gamma^*_{y,x}(\nu,\tau) = \Gamma_{f,y}(f,t)\Gamma^*_{g,x}(f,t). \qquad (9)$$

It is worth noting that $\Gamma_{f,g}(\nu,\tau)$ is an ambiguity function of the local delay and doppler variables τ and ν, and $\Gamma_{f,y}(f,t)$ is an ambiguity function of the global frequency and time variables f and t. It is also worth noting that from (9)

$$(\Gamma_{f,y}\Gamma^*_{g,x})(0,0) = \int\int(\Gamma_{f,g}\Gamma^*_{y,x})(\nu,\tau)\,d\nu d\tau;$$

i.e.,

$$(\Gamma_{f,y}\Gamma^*_{g,x})(0,0) = \langle f,y\rangle\langle g,x\rangle^* = \langle\Gamma_{f,g},\Gamma_{y,x}\rangle,$$

so that Moyal's formula can be viewed as a special case of the Sussman identity.

4. TIME-FREQUENCY ESTIMATION FOUR-CORNERS DIAGRAM

We consider the two-channel Rihaczek-based TF (time-frequency) estimate

$$P_e(t,f) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e(t-t', f-f')\gamma_{f,g}(t',f')\,dt'\,df' \quad (10)$$

in the East of Figure 3, where $\gamma_{f,g}(t,f)$ is the Rihaczek TF distribution of (1). In Figure 3, $*_i$ denotes convolution with respect to the ith variable, and $*$ denotes convolution with respect to both variables.

With the aid of Figure 1 and well known properties of Fourier transforms, the remaining corners are easily verified. We may interpret the TF estimate in the West as a window $w(\nu,\tau)$ applied to the signal cross-ambiguity function $\Gamma_{f,g}(\nu,\tau)$.

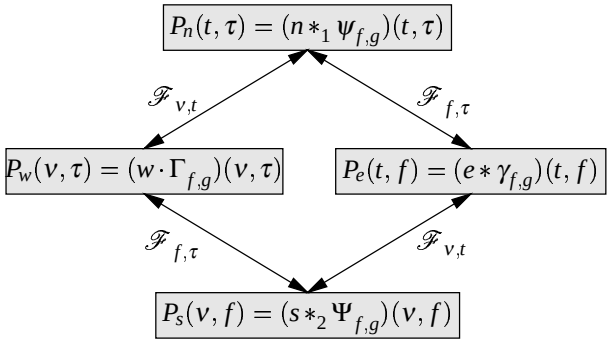


Figure 3: Rihaczek-based four-corners diagram for TF estimation.

In [6] a discrete-time version of

$$P(t,f) = \int\int f(t_1+t)g^*(t_2+t)Q(t_1,t_2)e^{j2\pi f(t_2-t_1)}\,dt_1 dt_2 \quad (11)$$

is considered and shown to be another representation for the most general member of Cohen's class [1] of TF distributions, which are quadratic, time- and frequency-translation invariant. Making the change of variables $t_1 = \tau - t'$ and $t_2 = -t'$ in (11) we obtain

$$\begin{aligned} P(t,f) &= \int\int Q(\tau-t',-t')\psi_{f,g}(t-t',\tau)e^{-j2\pi f\tau}\,dt'\,d\tau \\ &= \mathscr{F}_{f,\tau}(n *_1 \psi_{f,g})(t,\tau) = (e*\gamma_{f,g})(t,f) = P_e(t,f), \end{aligned}$$

where we have used $n(t,\tau) = Q(\tau-t,-t)$, so that $Q(t_1,t_2) = n(-t_2,t_1-t_2)$. It follows that the most general form of TF estimate that is quadratic and (t,f) translation invariant can be expressed in the form of $P_e(t,f)$ in (10).

Using Figure 3, we can write

$$P_e(t,f) = \int\int \Psi_{f,g}(\nu,f-f')s(\nu,f')e^{j2\pi\nu t}\,df'\,d\nu.$$

Letting $\nu = f_2 - f_1$ and $f' = f_1$, we obtain

$$P_e(t,f) = \iint F(f-f_1)G^*(f-f_2)\hat{Q}(f_1,f_2)e^{j2\pi(f_2-f_1)t}\,df_1\,df_2, \quad (12)$$

where we have used $\hat{Q}(f_1,f_2) = \mathscr{F}_{-f_2,t_2}\mathscr{F}_{f_1,t_1}Q(t_1,t_2) = \mathscr{F}_{-f_2,t_2}e(-t_2,f_1)e^{j2\pi f_1 t_2} = s(f_2-f_1,f_1)$, or $s(\nu,f) = \hat{Q}(f,\nu+f)$. This is a frequency-frequency smoothed version of (10), and the dual of (11). Thus, (10), (11), and (12) are t-f, t-t, and f-f representations for the Cohen class.

5. INTERFEROGRAMS AND SPECTROGRAMS

In Figure 3, if the West window $w(\nu,\tau) = \Gamma^*_{v_1,v_2}(\nu,\tau)$, is itself an ambiguity function involving two one-dimensional window functions $v_1(\cdot)$ and $v_2(\cdot)$, then using (9) the TF estimate becomes

$$P_e(t,f) = \Gamma_{f,v_1}(f,t)\Gamma^*_{g,v_2}(f,t). \quad (13)$$

If $\Gamma_{v_1,v_2}(\nu,\tau)$ depends on some parameter Θ, say $\Gamma_{v_1,v_2}(\nu,\tau) = \Gamma_{v_1,v_2}(\nu,\tau;\Theta)$, then we may consider a West window as a linear combination of the form

$$w(\nu,\tau) = \int \Gamma^*_{v_1v_2}(\nu,\tau;\Theta)W(\Theta)\,d\Theta \quad (14)$$

for a continuous parameter space, or

$$w(\nu,\tau) = \sum_i \Gamma^*_{v_1v_2}(\nu,\tau;\Theta_i)W(\Theta_i) \quad (15)$$

for a discrete parameter space, where $W(\Theta)$ is a weighting function.

If $v_1(t) = v_2(t) = M_{-f_o}v(t)$ and $W(\Theta) = V_o(f_o)$, then

$$w(\nu,\tau) = \int_{-\infty}^{\infty} \Gamma^*_{v,v}(\nu,\tau)V_o(f_o)e^{j2\pi f_o\tau}\,df_o,$$

corresponding in the North to $n(t,\tau) = \psi^*_{v,v}(-t,\tau)v_o(\tau)$, or $Q(t_1,t_2) = v^*(t_1)v_o(t_1-t_2)v(t_2)$, which is a dTd (diagonal-Toeplitz-diagonal) factorization of $Q(t_1,t_2)$. The resulting TF estimate is the weighted frequency-averaged spectrogram

$$P_e(t,f) = \int \Gamma_{f,v}(f-f_o,t)\Gamma^*_{g,v}(f-f_o,t)V_o(f_o)\,df_o.$$

If $v_1(t) = v_2(t) = T_{-t_o}v(t)$ and $W(\Theta) = v_o(t_o)$, then

$$w(\nu,\tau) = \int_{-\infty}^{\infty} \Gamma^*_{v,v}(\nu,\tau)v_o(t_o)e^{-j2\pi\nu t_o}\,dt_o,$$

corresponding in the South to $s(\nu,f) = \Psi^*_{v,v}(\nu,-f)V_o(\nu)$, or $\hat{Q}(f_1,f_2) = V^*(-f_1)V_o(f_2-f_1)V(-f_2)$, which is a dTd factorization of $\hat{Q}(f_1,f_2)$. The resulting TF estimate is the weighted time-averaged spectrogram

$$P_e(t,f) = \int \Gamma_{f,v}(f,t-t_o)\Gamma^*_{g,v}(f,t-t_o)v_o(t_o)\,dt_o.$$

If $v_1(t) = M_{f_o/2}v(t)$, $v_2(t) = M_{-f_o/2}v(t)$, and $W(\Theta) = V_o(f_o)$, then

$$w(\nu,\tau) = \int_{-\infty}^{\infty} \Gamma^*_{v,v}(\nu-f_o,\tau)V_o(f_o)e^{-j2\pi\frac{f_o}{2}\tau}\,df_o,$$

corresponding in the North to $n(t,\tau) = \psi^*_{v,v}(-t,\tau)v_o(t - \frac{\tau}{2})$, or $Q(t_1,t_2) = v^*(t_1)v_o(-\frac{t_1+t_2}{2})v(t_2)$, which is a dHd (diagonal-Hankel-diagonal) factorization of $Q(t_1,t_2)$. The resulting TF estimate is the weighted frequency-averaged interferogram

$$P_e(t,f) = \int \Gamma_{f,v}(f + \frac{f_o}{2},t)\Gamma^*_{g,v}(f - \frac{f_o}{2},t)V_o(f_o)\,df_o.$$

If $v_1(t) = T_{\frac{t_o}{2}}v(t)$, $v_2(t) = T_{-\frac{t_o}{2}}v(t)$ and $W(\Theta) = v_o(t_o)$, then

$$w(\nu,\tau) = \int_{-\infty}^{\infty} \Gamma^*_{v,v}(\nu,\tau - t_o)v_o(t_o)e^{-j2\pi\nu\frac{t_o}{2}}\,dt_o,$$

corresponding in the South to $s(\nu,f) = \Psi^*_{v,v}(\nu,-f)V_o(f + \frac{\nu}{2})$, or $\hat{Q}(f_1,f_2) = V^*(-f_1)V_o(\frac{f_1+f_2}{2})V(-f_2)$, which is a dHd factorization of $\hat{Q}(f_1,f_2)$. The resulting TF estimate is the weighted time-averaged Wigner-Ville distribution

$$P_e(t,f) = \int \Gamma_{f,v}(f,t + \frac{t_o}{2})\Gamma^*_{g,v}(f,t - \frac{t_o}{2})e^{-j2\pi f t_o}v_o(t_o)\,dt_o.$$

6. TIME-VARYING LINEAR SYSTEMS FOUR-CORNERS DIAGRAM

Consider a linear time-varying system with input delay-spread function $h(t,\tau)$ (the North in Figure 4), input $x(t)$ and output $y(t)$. The input-output relationship [7] is

$$y(t) = \int_{-\infty}^{\infty} h(t,\tau)x(t-\tau)\,d\tau \qquad (16)$$

in terms of the input delay-spread function, and

$$y(t) = \int_{-\infty}^{\infty} H(t,f)X(f)e^{j2\pi ft}\,df \qquad (17)$$

in terms of the time-varying frequency response $H(t,f)$. Standard Fourier transform identities may be used to fill out Figure 4 and write input-output equations using the input delay-doppler spread function $\sigma(\nu,\tau)$ or the output doppler-spread function $B(\nu,f)$.

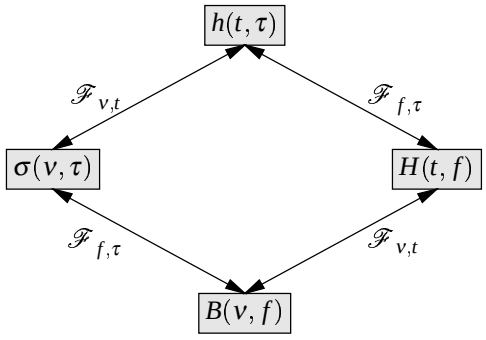


Figure 4: Basic input-output characterizations for LTV system.

7. WIDE-SENSE STATIONARY AND UNCORRELATED SCATTERING CASE

When the WSS and US assumptions are combined (yielding the WSSUS assumption), all of the two-dimensional Fourier transform relationships are reduced to one-dimensional Fourier transform relationships [7], as illustrated in Figure 5. It is important to note that the quantity $P_\sigma(\tau,\nu)$ in Figure 5 is commonly called the scattering function. It is also important to note that only the West corner of Figure 5,

$$R_H(\Delta f,\Delta t) = E(H(t+\Delta t,f)H^*(t,f-\Delta f)),$$

is a correlation function. The remaining corners are power densities, from which singular correlations may be constructed by applying δ-functions. In Figure 5, the global variables (τ,ν) play the same role as (t,f) previously, and the local variables $(\Delta f,\Delta t)$ play the same role as (ν,τ) previously.

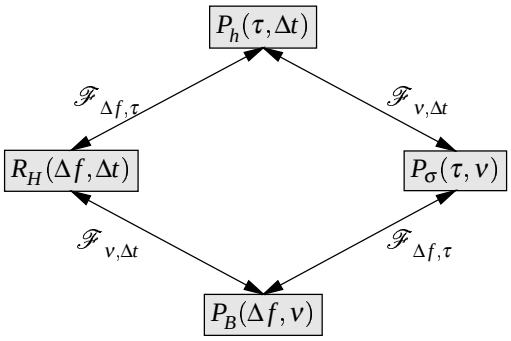


Figure 5: Four-corners diagram for WSSUS case.

8. CONNECTIONS BETWEEN TF DISTRIBUTIONS AND SCATTERING FUNCTION ESTIMATION

We consider a WSSUS channel with deterministic input signal $x(t)$, input delay-spread function $h(t,\tau)$ and output $y(t)$. The mean of the ambiguity function for y is

$$E(\Gamma_{y,y}(\Delta f,\Delta t)) = R_H(\Delta f,\Delta t)\Gamma_{x,x}(\Delta f,\Delta t), \qquad (18)$$

which is illustrated in Figure 6. Equation (18) is **the** fundamental result connecting input ambiguity to output ambiguity, through the time-frequency correlation R_H. Application of Fourier transform properties produces the remaining corners of Figure 6.

Comparing Figure 6 with Figure 3, we find that estimation of the scattering function (SF) P_σ is the same as estimation of the TF distribution except for the change in design rules: for TF we design $e(t,f)$ to estimate TF properties of signal $x(\cdot)$ ($f = g = x$ in Figure 3); for SF we design transmitted signal x to estimate P_σ. For TF we are trying to smooth whereas for SF we are trying to differentiate; or convolve for TF vs. deconvolve for SF.

Gaarder [2] proposed a two-stage translation-variant estimate for the scattering function using symmetric ambiguity functions as well as symmetric correlation functions. Here,

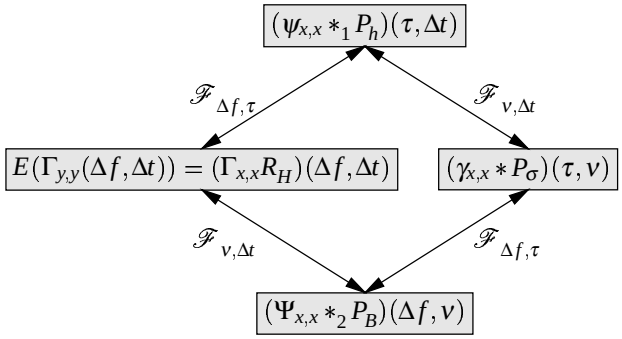


Figure 6: Rihaczek-based four-corners diagram for SF estimation.

we highlight a translation-invariant version of Gaarder's estimate with the notational conventions of the current paper. Stage 1 computes in the East

$$e(\tau,\nu) = |\Gamma_{y,h_1}(\tau,\nu)|^2,$$

which corresponds in the West (with the aid of the Sussman identity) to

$$w(\Delta f,\Delta t) = (\Gamma_{y,y}\cdot\Gamma^*_{h_1,h_1})(\Delta f,\Delta t).$$

Stage 2 adds a multiplication in the West by $h_2(\Delta f,\Delta t)$ to obtain

$$\hat{p}_\sigma(\Delta f,\Delta t) = (h_2\cdot\Gamma^*_{h_1,h_1}\cdot\Gamma_{y,y})(\Delta f,\Delta t),$$

corresponding in the East to the scattering function estimate

$$\hat{P}_\sigma(\tau,\nu) = (H_2 * |\Gamma_{y,h_1}(\tau,\nu)|^2)(\tau,\nu),$$

or

$$\hat{P}_\sigma(\tau,\nu) = (H_{\rm eq} * \gamma_{yy})(\tau,\nu),$$

where

$$h_{\rm eq}(\Delta f,\Delta t) = (h_2\cdot\Gamma^*_{h_1,h_1})(\Delta f,\Delta t).$$

We easily find that

$$E(\hat{p}_\sigma(\Delta f,\Delta t)) = (h_{\rm eq}\cdot\Gamma_{xx}\cdot R_H)(\Delta f,\Delta t),$$

or (letting $r(\Delta f,\Delta t) = (h_{\rm eq}\cdot\Gamma_{xx})(\Delta f,\Delta t)$)

$$E(\hat{P}_\sigma(\tau,\nu)) = (R * P_\sigma)(\tau,\nu).$$

9. CONCLUSION

We have presented a unified treatment of TFD estimation and SF estimation using a Rihaczek foundation. Four-corners diagrams are used to summarize key relationships. A fundamental key to the performance analysis of TFD and SF estimates is the Sussman four-corners diagram.

10. ACKNOWLEDGEMENT

This work was supported by the DARPA ISP program under contracts AFRL F33615- 02-C-1198 and FA9550-04-1-0371.

REFERENCES

[1] L. Cohen, "Time-Frequency Distributions–A Review," *Proc. IEEE*, vol. 77, No. 7, pp. 941-981, July 1989.

[2] N. T. Gaarder, "Scattering Function Estimation," *IEEE Trans. Information Theory*, vol. IT-14, No. 5, pp. 684-693, Sept. 1968.

[3] A. W. Rihaczek, "Signal Energy Distribution in Time and Frequency," *IEEE Trans. Information Theory*, vol. IT-14, pp. 369-374, May 1968.

[4] F. Hlawatsch, "Regularity and Unitarity of Bilinear Time-Frequency Signal Representations," *IEEE Trans. Information Theory*, vol. 38, No. 1, pp. 82-94, Jan. 1992.

[5] S. M. Sussman, "Least-Square Synthesis of Radar Ambiguity Functions," *IRE Trans. Information Theory*, vol. IT-8, pp. 246-254, April 1962.

[6] L. L. Scharf and B. Friedlander, "Toeplitz and Hankel Kernels for Estimating Time-Varying Spectra of Discrete-Time Random Processes," *IEEE Trans. Signal Processing*, vol. 49, pp. 179-189, Jan. 2001.

[7] P. A. Bello, "Characterization of randomly time-variant linear channels," *IEEE Trans. Communication Systems*," vol. CS-11, pp. 360-393, December 1963.

The Sussman, Moyal, and Janssen Formulas are Fourier Transform Consequences of a More Fundamental Identity

David C. Farden, *Member, IEEE*,
and Louis L. Scharf, *Fellow, IEEE*

***Abstract*— Janssen's formula is a sampled-data version of Moyal's, and both follow from Sussman's identity, which itself is a consequence of a more fundamental convolution identity.**

***Index Terms*— Sussman, Moyal, Janssen identities.**

I. Connections

Let $\psi_{fg}(t,\tau) = f(t+\tau)g^*(t)$ and define the adjoint $\tilde{\psi}_{fg}(t,\tau) = \psi^*_{fg}(-t,\tau)$. The fundamental convolution identity we shall exploit is this [1]:

$$(\psi_{fg} *_1 \tilde{\psi}_{yx})(t,\tau) = (\psi_{fy} *_1 \tilde{\psi}_{gx})(\tau,t), \tag{1}$$

where $*_1$ denotes convolution with respect to the first variable. The proof of (1) is a simple exercise in convolution. Note the swapping of g with y, and t with τ.

Define the ambiguity function

$$\Gamma_{fg}(\nu,\tau) = \mathcal{F}_{\nu,t}\psi_{fg}(t,\tau) = \int_{-\infty}^{\infty} f(t+\tau)g^*(t)e^{-j2\pi\nu t}\,dt,$$

and note that $\mathcal{F}_{\nu,t}\tilde{\psi}_{fg}(t,\tau) = \Gamma^*_{fg}(\nu,\tau)$. Now Fourier transform the LHS of (1) from t to ν, and the RHS from τ to f. Noting that each of these Fourier transforms is with respect to the first variable, the Fourier transform identities of Fig. 1 are readily obtained. In Fig. 1 we have defined $\Psi_{fg}(\nu,f) = \mathcal{F}_{f,\tau}\Gamma_{fg}(\nu,\tau)$, and $\tilde{\Psi}_{fg}(\nu,f) = \Psi^*_{fg}(\nu,-f)$.

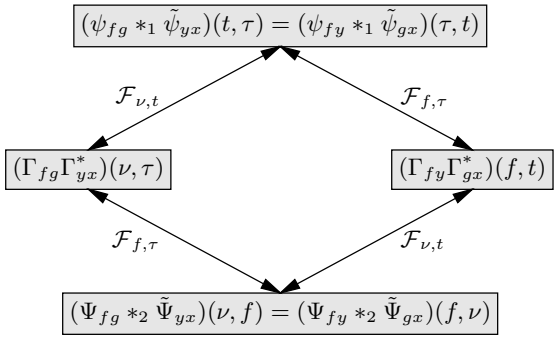


Fig. 1. Sussman four-corners diagram.

The two-dimensional Fourier transform pair

$$(\Gamma_{fg}\Gamma^*_{yx})(\nu,\tau) \Longleftrightarrow (\Gamma_{fy}\Gamma^*_{gx})(f,t), \tag{2}$$

known as the Sussman identity [2], follows from the fundamental convolution equality (1), together with the two one-dimensional Fourier transforms that separate the West and the East corners of Fig. 1 by a two-dimensional Fourier transform.

Now integrate the LHS of the Sussman identity (2), over (ν,τ), and denote it by the inner product notation $\langle\Gamma_{fg},\Gamma_{yx}\rangle$, to find

$$\langle\Gamma_{fg},\Gamma_{yx}\rangle = (\Gamma_{fy}\Gamma^*_{gx})(0,0),$$

which is just an initial value theorem of Fourier analysis. Then note that $\Gamma_{fy}(0,0) = \langle f,y\rangle$, meaning

$$\langle\Gamma_{fg},\Gamma_{yx}\rangle = \langle f,y\rangle\langle g,x\rangle^*. \tag{3}$$

D. Farden is with North Dakota State University, and L. Scharf is with Colorado State University.

This is Moyal's identity [3]. Thus Moyal's identity is a consequence of Sussman's identity, which in turn is a Fourier transform version of the more fundamental identity (1).

The sampled-data version of Sussman's identity is, by Poisson's sum formula,

$$(\Gamma_{fg}\Gamma^*_{yx})(mF,nT) \Longleftrightarrow \frac{1}{FT}\sum_{k,\ell}(\Gamma_{fy}\Gamma^*_{gx})(f+\frac{k}{T},t+\frac{\ell}{F}). \tag{4}$$

Invoke an initial-value theorem to write the sampled data version of Moyal's formula as

$$\sum_{m,n}(\Gamma_{fg}\Gamma^*_{yx})(mF,nT) = \frac{1}{FT}\sum_{k,\ell}(\Gamma_{fy}\Gamma^*_{gx})(\frac{k}{T},\frac{\ell}{F}),$$

or, in terms of discrete inner-products, as

$$\langle\Gamma_{fg},\Gamma_{yx}\rangle = \frac{1}{FT}\langle\Gamma_{fy},\Gamma_{gx}\rangle. \tag{5}$$

This actually a generalized version of Janssen's formula [4]. That is, when $g_{\nu,\tau}(t) = g(t-\tau)e^{j2\pi\nu t}$, then $\Gamma_{fg}(\nu,\tau) = \langle f,g_{\nu,\tau}\rangle e^{j2\pi\nu\tau}$, and we may write the sampled-data version of Moyal's formula (5) as

$$\sum_{m,n}\langle f,g_{mF,nT}\rangle\langle y,x_{mF,nT}\rangle^* = \frac{1}{FT}\sum_{k,\ell}\langle f,y_{\frac{k}{T},\frac{\ell}{F}}\rangle\langle g,x_{\frac{k}{T},\frac{\ell}{F}}\rangle^*, \tag{6}$$

which is the usual form of Janssen's identity [4].

II. Conclusion

Thus, the equivalent fundamental identities are (1) and (2), the latter called Sussman's identity, with Moyal's formula (3) following from an initial value theorem of Fourier analysis, and Janssen's equality (6) following from Poisson's sum formula and an initial value theorem. That is, Janssen's formula is a sampled-data version of Moyal's, and both follow from Sussman's identity, which itself is a consequence of the fundamental identity (1).

Acknowledgment

This work was supported by the DARPA ISP program under contracts AFRL F33615- 02-C-1198 and FA9550-04-1-0371.

References

[1] D. C. Farden and L. L. Scharf, "Estimating Scattering Functions and Time-Frequency Distributions using the Rihaczek Distribution," IEEE Workshop on Sensor Array and Multichannel Signal Processing (SAM), Sitges, Spain, July 18-21, 2004.

[2] S. M. Sussman, "Least-Square Synthesis of Radar Ambiguity Functions," *IRE Trans. Information Theory*, vol. IT-8, pp. 246-254, April 1962.

[3] T. A. Claasen and W. F. G. Mecklenbräuker, "The Wigner Distribution– A Tool for Time-Frequency Signal Analysis; Part I: Continuous-Time Signals," *Philips J. Research*, 35(3), pp. 217-250, 1980.

[4] A. J. E. M. Janssen, "Representations of Gabor Frame Operators," NATO-ASI 2000, II Ciocco, Tuscany (Italy), July 2-15, 2000.

Technical Report for Summer 2003
Agreement No: CSL-02394.01:JAG

Consultant: John A. Gubner

Contents

Introduction

The purpose of the ISP Program (Integrated Sensing and Processing) is to develop systems for optimal adaptive cooperation among distributed sensors and a global processing unit (GPU). This cooperation will adapt the amount of computation/processing done at local sensors, the communication bandwidth to the GPU, and the amount of computation/processing done at the GPU.

As discussed at the June 2003 ISP Kickoff Meeting, one important research component is the estimation of channel statistics. A prerequisite of this is a channel model. Hence, Section 1 of this report is devoted to the discussion of a multipath-Doppler channel model as would be encountered in a wireless environment. This is the time-varying communication environment over which the distributed sensors and GPU must communicate.

The GPU will be receiving signals from many sensors at the same time, and it will be necessary to isolate the signal from the desired sensor. Four different models for the channel seen by the desired user and the channels seen by interfering users are discussed in Section 2. Our contribution here is to show that, even when the interference is infinite dimensional, the analysis of case 2 can be reduced to case 1, and the analysis of case 3 can be reduced to case 4.

Section 3 is concerned with the blind estimation of the interference-free signal subspace, which naturally arises in the zero-forcing detector of subspace signals in subspace interference and noise. This section extends known finite-dimensional results to the infinite-dimensional case.

In the preceding sections, no assumption was made about the signaling waveforms of the desired user. Section 4 shows how to design these waveforms to maximize the average mutual information between the channel coefficients and the received waveform. It is shown that the optimum waveforms depend only on the covariance matrix of the channel coefficients.

Section 5 looks at GPU design when each sensor can transit only one bit of information about its measurement. This section also considers an approach to GPU design when each sensor is allowed to send a multi-bit word of information about its measurement. Further research along these lines should allow for an adaptive system in which the sensing/computation of the local sensors, the information they transmit to the GPU, and the processing at the GPU are adjusted based on time-varying channel capacity.

1. Multipath-Doppler Channel Models

To exploit the matched subspace detectors of [8], [9], we first give a concise derivation of the multipath-Doppler model of [7] in Section 1.1. Then in Section 1.2 we discuss orthogonal projections onto the multipath-Doppler subspace. Interestingly, the quantities appearing in the normal equations can be expressed in terms of the ambiguity function of the transmitted waveform and the cross-ambiguity function of the received waveform and the transmitted waveform. In Section 1.3, an alternative derivation yields a different model parameterization.

1.1. First Derivation

In this section we first derive a **sampling theorem** for the output of a linear, time-varying system when:

1. The input is bandlimited to W.
2. The output is of interest only during the finite time interval $[0,T]$.

The second step is to show that if the channel causal and has finite multipath spread τ_m and finite Doppler spread B_d, then the infinite series of our sampling theorem can be truncated with negligible error.

Consider a time-varying, linear channel model of the form

$$y(t) \;=\; \int h(t,\tau)x(t-\tau)\,d\tau. \tag{1.1}$$

Writing this convolution as the inverse transform of the transforms, we have

$$y(t) \;=\; \int H(t,f)X(f)e^{j2\pi ft}\,df,$$

where H is the **time-varying transfer function**

$$H(t,f) \;:=\; \int h(t,\tau)e^{-j2\pi f\tau}\,d\tau,$$

and

$$X(f) \;:=\; \int x(\tau)e^{-j2\pi f\tau}\,d\tau.$$

If we now use the fact that the signal x is bandlimited to W and that we are interested in $y(t)$ only for $0 \le t \le T$, then

$$y(t) \;=\; \int_{-W}^{W} H(t,f)X(f)e^{j2\pi ft}\,df, \quad 0 \le t \le T. \tag{1.2}$$

Since the foregoing expression involves $H(t,f)$ for $(t,f) \in [0,T]\times[-W,W]$, we can expand H in the **bivariate Fourier series**

$$H(t,f) \;=\; \sum_k \sum_\ell H_{k,\ell}\, e^{j2\pi kt/T} e^{-j2\pi \ell f/2W}.$$

Substituting this expansion into (1.2) yields, for $0 \le t \le T$,

$$\begin{aligned} y(t) &= \sum_k \sum_\ell H_{k,\ell}\, e^{j2\pi kt/T} \int_{-W}^{W} X(f) e^{j2\pi f(t-\ell/2W)}\,df \\ &= \sum_k \sum_\ell H_{k,\ell}\, e^{j2\pi kt/T} x(t-\ell/2W). \end{aligned} \tag{1.3}$$

To complete the sampling theorem, we now analyze the formulas for the coefficients $H_{k,\ell}$. Write

$$\begin{aligned} H_{k,\ell} &= \frac{1}{2WT}\int_0^T \int_{-W}^{W} H(t,f)e^{-j2\pi kt/T}e^{j2\pi \ell f/2W}\,df\,dt \\ &= \frac{1}{2WT}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} I_{[0,T]}(t)I_{[-W,W]}(f)H(t,f)e^{-j2\pi kt/T}e^{j2\pi \ell f/2W}\,df\,dt. \end{aligned}$$

Here $I_{[0,T]}(t)$ is the indicator function of the set $[0,T]$; i.e., $I_{[0,T]}(t) = 1$ for $t \in [0,T]$ and is zero otherwise. We regard each integral as the transform or inverse transform of a product. Thus,

$$H_{k,\ell} = \frac{1}{T}\int_{-\infty}^{\infty} I_{[0,T]}(t)e^{-j2\pi kt/T}\int_{-\infty}^{\infty} h(t,\tau)\,\mathrm{sinc}\Big(2W\Big[\frac{\ell}{2W}-\tau\Big]\Big)\,d\tau\,dt \tag{1.4}$$

$$= \int_{-\infty}^{\infty}\mathrm{sinc}\Big(2W\Big[\frac{\ell}{2W}-\tau\Big]\Big)\int_{-\infty}^{\infty} C(\nu,\tau)\,\mathrm{sinc}\Big(T\Big[\frac{k}{T}-\nu\Big]\Big)e^{-j\pi T(k/T-\nu)}\,d\nu\,d\tau, \tag{1.5}$$

where

$$C(\nu,\tau) := \int h(t,\tau)e^{-j2\pi\nu t}\,dt \tag{1.6}$$

is called the **scattering function**. Notice that if W and T are both large, then the sinc functions act like impulses, and

$$H_{k,\ell} \approx \frac{C(k/T,\ell/2W)}{2WT}.$$

Hence, we call the $H_{k,\ell}$ the **scattering coefficients**.

We now impose further assumptions. If the system in (1.1) is causal, then $h(t,\tau) = 0$ for $\tau < 0$, and so

$$y(t) = \int_0^{\infty} h(t,\tau)x(t-\tau)\,d\tau. \tag{1.7}$$

If the system's response to an impulse at time t_0 is finite duration, e.g., zero for $t > t_0 + \tau_m$, then

$$h(t,\tau) = 0, \quad \text{for } \tau > \tau_m,$$

and (1.7) becomes

$$y(t) = \int_0^{\tau_m} h(t,\tau)x(t-\tau)\,d\tau.$$

Furthermore, the inner integral in (1.4) becomes

$$\int_0^{\tau_m} h(t,\tau)\,\mathrm{sinc}\Big(2W\Big[\frac{\ell}{2W}-\tau\Big]\Big)\,d\tau,$$

which is approximately zero[1] for $\ell \le -1$ and for $\ell \ge 2W\tau_m + 1$. Taking $L := \lceil 2W\tau_m \rceil$, we have $H_{k,\ell} \approx 0$ for $\ell < 0$ and $\ell > L$. We next assume that the channel has finite Doppler spread B_d; i.e.,

$$C(\nu,\tau) = 0, \quad \text{for } |\nu| > B_d.$$

[1]The convolution is significant when the main lobe of the sinc is touching $[0,T]$.

Then the inner integral in (1.5) becomes

$$\int_{-B_d}^{B_d} C(\nu,\tau)\,\mathrm{sinc}\Big(T\Big[\frac{k}{T}-\nu\Big]\Big)e^{-j\pi T(k/T-\nu)}\,d\nu.$$

This convolution is approximately zero for $|k| > B_dT + 1$. We therefore put $M := \lceil B_dT \rceil$ so that $H_{k,\ell} \approx 0$ for $|k| > M$. We now have the approximation of (1.3),

$$y(t) \;\approx\; \sum_{k=-M}^{M}\sum_{\ell=0}^{L} H_{k,\ell}\,e^{j2\pi kt/T}x(t-\ell/2W), \quad 0 \le t \le T.$$

The importance of this representation is that the only unknowns are the scattering coefficients $H_{k,\ell}$. The values of M, L, T, and W are known. This is to be compared with the usual representation

$$y(t) \;\approx\; \sum_{k}\sum_{\ell} H_{k,\ell}\,e^{j2\pi\nu_k t}x(t-\tau_\ell), \quad 0 \le t \le T,$$

where the Doppler frequencies ν_k and delays τ_ℓ are also unknown. Note, however, that the number of terms k and ℓ in this representation will typically be less than $2M+1$ and $L+1$, respectively. In other words, the linear model reduces complexity by increasing the number of coefficients.

1.2. Projections onto the Signal Subspace

Based on the foregoing analysis, when the bandlimited waveform x is sent over the multipath-Doppler channel, we model the received waveform by

$$y(t) \;=\; \sum_{k=-M}^{M}\sum_{\ell=0}^{L} H_{k,\ell}\,e^{j2\pi kt/T}x(t-\ell/2W) + n(t), \quad 0 \le t \le T,$$

where $n(t)$ is noise. Consider the problem of projecting y onto the subspace

$$\mathcal{S} \;:=\; \mathrm{span}\{s_{k,\ell}, -M \le k \le M, 0 \le \ell \le L\},$$

where

$$s_{k,\ell}(t) \;:=\; e^{j2\pi kt/T}x(t-\ell/2W), \quad 0 \le t \le T.$$

The usual orthogonality-principle argument [5, §3.6] says that the projection $\widehat{y}$ is given by

$$\widehat{y}(t) \;=\; \sum_{k=-M}^{M}\sum_{\ell=0}^{L} \widehat{H}_{k,\ell}\,s_{k,\ell}(t),$$

where the coefficients $\widehat{H}_{k,\ell}$ solve the normal equations. The entries of the Gram matrix are given by $\langle s_{k,\ell}, s_{k',\ell'}\rangle$, and the components of the "other side" of the normal equations are given by

$$\begin{aligned}
\langle y, s_{k',\ell'}\rangle &= \int y(t)s_{k',\ell'}(t)^*\,dt \\
&= \int y(t)e^{-j2\pi k't/T}x(t-\ell'/2W)^*\,dt \\
&= \int y(\theta+\ell'/2W)x(\theta)^*e^{-j2\pi k'(\theta+\ell'/2W)/T}\,d\theta \\
&= e^{-j2\pi k'\ell'/2WT}A_{yx}(k'/T,\ell'/2W),
\end{aligned}$$

where the $*$ indicates complex conjugation, and A_{yx} is the **cross-ambiguity function**,

$$A_{yx}(\nu,\tau) \;:=\; \int y(t+\tau)x(t)^* e^{-j2\pi\nu t}\,dt.$$

In the foregoing, it is understood that $y(t) = 0$ for t outside $[0,T]$. The entries of the Gram matrix are

$$\langle s_{k,\ell}, s_{k',\ell'}\rangle \;=\; \int e^{j2\pi kt/T} x(t-\ell/2W) e^{-j2\pi k't/T} x(t-\ell'/2W)^*\,dt,$$

where the range of integration is $[0,T]$. If the temporal support of all the $x(t-\ell/2W)$ is *essentially* contained in $[0,T]$, we can write[2]

$$\begin{aligned}\langle s_{k,\ell}, s_{k',\ell'}\rangle &= \int x(\theta+[\ell'-\ell]/2W)x(\theta)^* e^{j2\pi[k-k'](\theta+\ell'/2W)/T}\,d\theta \\ &= e^{-j2\pi[k'-k]\ell'/2WT}\int x(\theta+[\ell'-\ell]/2W)x(\theta)^* e^{-j2\pi[k'-k]\theta/T}\,d\theta \qquad (1.8)\\ &= e^{-j2\pi[k'-k]\ell'/2WT} A_{xx}([k'-k]/T,[\ell'-\ell]/2W).\end{aligned}$$

If the assumption about the temporal support of the $x(t-\ell/2W)$ is not valid, then the integral in (1.8) is replaced by

$$\int_{-\ell'/2W}^{T-\ell'/2W} x(\theta+[\ell'-\ell]/2W)x(\theta)^* e^{-j2\pi[k'-k]\theta/T}\,d\theta.$$

1.3. Derivation of the Second Model

The derivation in Section 1.1 was based on the bivariate Fourier expansion of $H(t,f)$ followed by the finite multipath delay and finite Doppler spread assumptions. Alternatively, we can rewrite (1.1) in terms of $C(\nu,\tau)$ defined in (1.6). Thus,

$$y(t) \;=\; \int\Big[\int C(\nu,\tau)e^{j2\pi\nu t}\,d\nu\Big]x(t-\tau)\,d\tau.$$

If we now assume that the system is causal with finite multipath spread τ_m and finite Doppler spread B_d, we have

$$y(t) \;=\; \int_0^{\tau_m}\Big[\int_{-B_d}^{B_d} C(\nu,\tau)e^{j2\pi\nu t}\,d\nu\Big]x(t-\tau)\,d\tau. \qquad (1.9)$$

This time we expand $C(\nu,\tau)$ in a bivariate Fourier series on $[-B_d,B_d]\times[0,\tau_m]$. Substituting

$$C(\nu,\tau) \;=\; \sum_\ell\sum_k C_{\ell,k}\, e^{-j2\pi\ell\nu/2B_d} e^{j2\pi k\tau/\tau_m}$$

into (1.9) yields

$$\begin{aligned} y(t) &= \sum_\ell\sum_k C_{\ell,k}\int_0^{\tau_m}\Big[\int_{-B_d}^{B_d} e^{j2\pi\nu(t-\ell/2B_d)}\,d\nu\Big]x(t-\tau)e^{j2\pi k\tau/\tau_m}\,d\tau \\ &= 2B_d\sum_\ell\sum_k C_{\ell,k}\,\mathrm{sinc}\Big(2B_d\Big[t-\frac{\ell}{2B_d}\Big]\Big)\int_0^{\tau_m} x(t-\tau)e^{j2\pi k\tau/\tau_m}\,d\tau.\end{aligned}$$

[2]Since the nonzero waveform x is bandlimited, it cannot be time limited.

In this last integral, substitute

$$x(t-\tau) \;=\; \int X(f)e^{j2\pi f(t-\tau)}\,df$$

to get

$$\begin{aligned} y(t) &= 2B_d\tau_m \sum_\ell \sum_k C_{\ell,k}\,\mathrm{sinc}\Big(2B_d\Big[t-\frac{\ell}{2B_d}\Big]\Big) \\ &\quad \cdot \int X(f)e^{-j\pi\tau_m(f-k/\tau_m)}\,\mathrm{sinc}(\tau_m[f-k/\tau_m])e^{j2\pi ft}\,df \\ &= 2B_d\tau_m \sum_\ell \sum_k C_{\ell,k}\,(-1)^k\,\mathrm{sinc}\Big(2B_d\Big[t-\frac{\ell}{2B_d}\Big]\Big) \\ &\quad \cdot \int X(f)\,\mathrm{sinc}(\tau_m[f-k/\tau_m])e^{j2\pi f(t-\tau_m/2)}\,df. \end{aligned} \tag{1.10}$$

For τ_m large, the last sinc function acts like a delta function and so

$$\begin{aligned} y(t) &\approx 2B_d \sum_\ell \sum_k C_{\ell,k}\,(-1)^k\,\mathrm{sinc}\Big(2B_d\Big[t-\frac{\ell}{2B_d}\Big]\Big)X(k/\tau_m)e^{j2\pi(k/\tau_m)(t-\tau_m/2)} \\ &= 2B_d \sum_\ell \sum_k C_{\ell,k}\,\mathrm{sinc}\Big(2B_d\Big[t-\frac{\ell}{2B_d}\Big]\Big)X(k/\tau_m)e^{j2\pi kt/\tau_m}. \end{aligned}$$

We now turn to the coefficients $C_{\ell,k}$. Write

$$\begin{aligned} C_{\ell,k} &= \frac{1}{2B_d\tau_m}\int_0^{\tau_m}\int_{-B_d}^{B_d} C(\nu,\tau)e^{j2\pi\ell\nu/2B_d}e^{-j2\pi k\tau/\tau_m}\,d\nu\,d\tau \\ &= \frac{1}{2B_d\tau_m}\int_0^{\tau_m}\Big[\int I_{[-B_d,B_d]}(\nu)C(\nu,\tau)e^{j2\pi\ell\nu/2B_d}\,d\nu\Big]e^{-j2\pi k\tau/\tau_m}\,d\tau \\ &= \frac{1}{\tau_m}\int_0^{\tau_m}\Big[\int \mathrm{sinc}\Big(2B_d\Big[\frac{\ell}{2B_d}-t\Big]\Big)h(t,\tau)\,dt\Big]e^{-j2\pi k\tau/\tau_m}\,d\tau \\ &= \int \mathrm{sinc}\Big(2B_d\Big[\frac{\ell}{2B_d}-t\Big]\Big)\Big[\frac{1}{\tau_m}\int I_{[0,\tau_m]}(\tau)h(t,\tau)e^{-j2\pi k\tau/\tau_m}\,d\tau\Big]\,dt \\ &= \int \mathrm{sinc}\Big(2B_d\Big[\frac{\ell}{2B_d}-t\Big]\Big)\Big[\int H(t,f)\,\mathrm{sinc}\Big(\tau_m\Big[\frac{k}{\tau_m}-f\Big]\Big)e^{-j\pi\tau_m(k/\tau_m-f)}\,df\Big]dt. \end{aligned} \tag{1.11}$$

Notice that for large τ_m and B_d, the sinc functions act like impulses, and so

$$C_{\ell,k} \;\approx\; \frac{H(\ell/2B_d, k/\tau_m)}{2B_d\tau_m}.$$

For this reason, we call the $C_{\ell,k}$ the **channel coefficients**. If we assume that the time-varying transfer function is bandlimited to W, then (1.11) tells us that $C_{\ell,k} \approx 0$ for $|k| > \lceil W\tau_m\rceil$. Similarly, if we restrict t to $[0,T]$ in (1.10), we see that for $\ell < 0$ and $\ell > \lceil 2B_dT\rceil$, $C_{\ell,k} \approx 0$. Thus, we can approximate $y(t)$ by using finite sums in (1.10). So far, we do not see how to exploit this representation.

2. Subspace Signals in Subspace Interference and Noise

Consider a receiver whose input is

$$y \;=\; a + b + n,$$

where a is the desired signal, b is an interference signal, and n is white noise. It is assumed that a belongs to a finite-dimensional subspace $\mathcal{A}$ and b belongs to a subspace $\mathcal{B}$, which may be infinite dimensional. There are four (Gaussian) cases to consider [10]:

1. a and b are deterministic but unknown.
2. a is deterministic and unknown, but b is random and independent of n.
3. b is deterministic and unknown, but a is random and independent of n.
4. a and b are random with a, b, and n independent.

According to [10, p. 2939], all except case 3 have been discussed at length in the literature when a, b, and n are finite-dimensional vectors in $\mathbb{C}^n$. Case 3 (for finite-dimensional vectors) is treated in [10, Section IV-C]. Here we first generalize by allowing the interference to lie in an infinite-dimensional subspace. We then show that case 2 can be transformed into case 1 with no interference term, and we show that case 3 can be transformed into case 4 with no interference term. Hence, once we know how to solve cases 1 and 4, those solutions can be used to solve cases 2 and 3, respectively. The importance of these transformation is that case 4 is well understood, and case 1 has recently been solved for constraints on a and with $\mathcal{B}$ being infinite dimensional [3].

2.1. Model Details

We take a to be of the form

$$a \;=\; Au \;:=\; \sum_{k=1}^{p} u_k a_k,$$

where $a_1, \ldots, a_p$ are linearly independent. Thus, a lives in the finite-dimensional subspace

$$\mathcal{A} \;:=\; \operatorname{span}\{a_1, \ldots, a_p\},$$

and $u := [u_1, \ldots, u_p]'$ is either deterministic and unknown or random with zero mean and covariance matrix R_u. When b is deterministic, it is allowed to lie, for example, in a closed, infinite dimensional subspace $\mathcal{B}$ of $L^2[0,T]$. When b is a zero-mean random process with covariance function $r_b(t_1,t_2)$, the process is assumed to have a Karhunen–Loève expansion, e.g.,

$$b(t) \;=\; \sum_{k=1}^{\infty} B_k s_k(t),$$

where

$$B_k \;=\; \int_0^T b(t) s_k(t)^*\, dt, \tag{2.1}$$

the s_k are orthonormal, the B_k are independent, and $\int_0^T r_b(t,\tau) s_k(\tau)\, d\tau = \beta_k s_k(t)$. Here we take

$$\mathcal{B} \;=\; \overline{\operatorname{span}\{s_k\}_{k=1}^{\infty}}.$$

We have recently solved case 1 for infinite-dimensional $\mathcal{B}$ in [3]. In case 4, linear estimation of a (or u) is a standard Wiener-filter problem. If Karhunen–Loève expansions are valid, then detection is also straightforward. We next show that case 2 can be reduced to a special version of case 1 in which $\mathcal{B}$ is the zero subspace. Similarly, we show that case 3 can be reduced to a special version of case 4 in which there is no interference.

2.2. Reduction of Case 2 to Case 1

The first step is to write $y = a + b + n$ as $y = a + w$, where $w := b + n$. The covariance function of w is

$$r_w(t_1, t_2) \;=\; r_b(t_1, t_2) + \sigma^2 \delta(t_1 - t_2),$$

where δ is the Dirac delta function. The corresponding covariance operator is $R_w := R_b + \sigma^2 I$, where I is the identity operator, and

$$(R_b x)(t) \;:=\; \int_0^T r_b(t, \tau) x(\tau)\, d\tau.$$

The basic idea is that observing y is equivalent to observing

$$\check{y} \;:=\; R_w^{-1/2} y \;=\; \check{a} + \check{n},$$

where $\check{a} := R_w^{-1/2} a$ and $\check{n} := R_w^{-1/2} w$. Since $\check{n}$ is white noise, and since $\check{a} = \check{A} u$, where $\check{A} := R_w^{-1/2} A$, we see that $\check{y} = \check{A} u + \check{n}$ is exactly case 1 with no interference.

It remains to check a few details such as the existence of $R_w^{-1/2}$ and the fact that $\check{n}$ is white noise.

To find determine $R_w^{-1/2}$, we proceed as follows. Assuming that

$$\int_0^T \int_0^T |r_b(t, \tau)|^2 \, dt \, d\tau \;<\; \infty,$$

it follows that R_b is a compact operator [2, pp. 86–87].[3] By the spectral theorem for compact, self-adjoint operators [2, p. 113], R_b has the representation

$$R_b x \;=\; \sum_{k=1}^{\infty} \beta_k \langle x, s_k \rangle s_k,$$

where the β_k are positive, nonincreasing eigenvalues with corresponding orthonormal eigenvectors s_k; i.e., $R_b s_k = \beta_k s_k$. Furthermore, we have the *orthogonal* direct sum decomposition [2, p. 115]

$$L^2[0, T] \;=\; \ker R_b \oplus \mathcal{B},$$

where $\mathcal{B}$ was defined above. For $x \in L^2[0, T]$, we can always write $x = \hat{x} + \tilde{x}$, where $\hat{x}$ is the projection of x onto $\ker R_b$, and

$$\tilde{x} \;=\; \sum_{k=1}^{\infty} \langle x, s_k \rangle s_k$$

[3] To guarantee the existence of the Karhunen–Loève expansion, we need r_b to be continuous. Note that continuity of r_b is enough to make R_b compact. Continuity of r_b is assured if we assume $b(t)$ is mean-square continuous.

is the projection of x onto $\mathcal{B}$. Thus,

$$\begin{aligned} R_w x &= (\sigma^2 I + R_b)x \\ &= \sigma^2 x + \sum_{k=1}^{\infty} \beta_k \langle x, s_k \rangle s_k \\ &= \sigma^2(\hat{x} + \tilde{x}) + \sum_{k=1}^{\infty} \beta_k \langle x, s_k \rangle s_k \\ &= \sigma^2 \hat{x} + \sum_{k=1}^{\infty} (\sigma^2 + \beta_k) \langle x, s_k \rangle s_k. \end{aligned} \tag{2.2}$$

This suggests that

$$R_w^{1/2} x := \sigma \hat{x} + \sum_{k=1}^{\infty} (\sigma^2 + \beta_k)^{1/2} \langle x, s_k \rangle s_k \tag{2.3}$$

should be self-adjoint and satisfy $R_w^{1/2}(R_w^{1/2} x) = R_w x$. This is easily checked to be the case. Next, it is an easy exercise to check that

$$R_w^{-1} y = \sigma^{-2} y + \sum_{k=1}^{\infty} \frac{-\beta_k/\sigma^2}{\sigma^2 + \beta_k} \langle y, s_k \rangle s_k.$$

Then, just as (2.2) led to (2.3), we easily find that

$$R_w^{-1/2} y = \sigma^{-1} \hat{y} + \sum_{k=1}^{\infty} (\sigma^2 + \beta_k)^{-1/2} \langle y, s_k \rangle s_k, \tag{2.4}$$

where $\hat{y}$ is the projection of y onto $\ker R_b$.

It remains to show that $\check{n} := R_w^{-1/2} w$ is white noise. Write

$$\begin{aligned} \check{n}(t) &= \sigma^{-1} \hat{w}(t) + \sum_{k=1}^{\infty} (\sigma^2 + \beta_k)^{-1/2} \langle w, s_k \rangle s_k(t) \\ &= \sigma^{-1} \hat{n}(t) + \sum_{k=1}^{\infty} (\sigma^2 + \beta_k)^{-1/2} (B_k + N_k) s_k(t), \end{aligned}$$

where B_k was defined in (2.1), N_k is defined similarly, and

$$\hat{n}(t) = n(t) - \sum_{k=1}^{\infty} N_k s_k(t).$$

Using the formula

$$\mathsf{E}[\hat{n}(t_1)\hat{n}(t_2)^*] = \sigma^2 \delta(t_1 - t_2) - \sigma^2 \sum_{k=1}^{\infty} s_k(t_1) s_k(t_2)^*$$

along with $\mathsf{E}[B_k N_l^*] = 0$, $\mathsf{E}[B_k B_l^*] = \beta_k \delta_{kl}$, and $\mathsf{E}[N_k N_l^*] = \sigma^2 \delta_{kl}$, it is straightforward to verify that $\mathsf{E}[\check{n}(t_1)\check{n}(t_2)^*] = \delta(t_1 - t_2)$. Hence, $\check{n}$ is white noise.

2.3. Reduction of Case 3 to Case 4

Let $P_{\mathcal{B}}a$ denote the projection of a onto $\mathcal{B}$, and let $P_{\mathcal{B}}^{\perp}a$ denote the projection of a onto $\mathcal{B}^{\perp}$. Then $y = a + b + n$ can be written as

$$y \;=\; P_{\mathcal{B}}^{\perp}a + \tilde{b} + n,$$

where $\tilde{b} := P_{\mathcal{B}}a + b$. In case 3, b is an unknown element of $\mathcal{B}$, and so our model is equivalent to

$$y \;=\; P_{\mathcal{B}}^{\perp}a + b + n,$$

where b is just another unknown element of $\mathcal{B}$. Put $\mathcal{G} := P_{\mathcal{B}}^{\perp}(\mathcal{A})$. Since $\mathcal{G} \subset \mathcal{B}^{\perp}$, $P_{\mathcal{G}}b = 0$. Since $a \in \mathcal{A}$, $P_{\mathcal{B}}^{\perp}a \in \mathcal{G}$, and so $P_{\mathcal{G}}P_{\mathcal{B}}^{\perp}a = P_{\mathcal{B}}^{\perp}a$. Hence,

$$z \;:=\; P_{\mathcal{G}}y \;=\; P_{\mathcal{B}}^{\perp}a + v,$$

where $v := P_{\mathcal{G}}n$. Notice that z and $y - z = b + P_{\mathcal{G}}^{\perp}n$ are uncorrelated and therefore independent. Also, $y - z$ is independent of a and of v. Hence, there is no loss of information about a if we work with z instead of y. To conclude, note that since $\mathcal{A}$ is finite dimensional, so is $\mathcal{G}$. Hence, instead of working with z, we can work with its coordinate vector relative to some orthonormal basis of $\mathcal{G}$. Denote this coordinate vector by $\underline{z}$ and similarly for $\underline{v}$. Since the covariance matrix of $\underline{v}$ is $\sigma^2 I$, $\underline{z}$ is a version of case 4.

Remark. If n is Gaussian, and if whenever a and/or b are random they are also Gaussian, then since all transformations here are linear, Gaussianity is preserved.

3. Blind Identification of the Interference-Free Signal Subspace and the Signal Coefficient Covariance

Consider a signal detection problem in which the received waveform is

$$y(t) \;=\; a(t) + b(t) + n(t), \quad 0 \le t \le T,$$

where a is the random signal to be detected, b is a random interference process, and n is a white noise process. If a belongs to a subspace $\mathcal{A}$, and b belongs to a subspace $\mathcal{B}$, then a key step in designing suboptimal linear detectors for CDMA systems [12], and more generally, matched subspace detectors [8], [9], is the characterization of the subspace $P_{\mathcal{B}}^{\perp}(\mathcal{A})$ that results from projecting the elements of $\mathcal{A}$ onto the orthogonal complement of $\mathcal{B}$. The problem of finding $P_{\mathcal{B}}^{\perp}(\mathcal{A})$ when $\mathcal{B}$ is unknown was studied by Scharf and McCloud [10] when a, b, and n were finite-dimensional random vectors. Here we allow a, b, and n to be waveforms, and we specifically allow $\mathcal{B}$ to be infinite dimensional.

3.1. System Model

We assume that the signal is of the form

$$a(t) \;=\; (Au)(t) \;:=\; \sum_{k=1}^{p} u_k a_k(t), \tag{3.1}$$

where $a_1, \ldots, a_p$ are linearly independent waveforms in $L^2[0,T]$, and $u := [u_1, \ldots, u_p]'$ is a random vector in $\mathbb{C}^p$. In other words, the operator A takes the random column vector u and returns the random waveform Au, which lives in the p-dimensional subspace of waveforms

$$\mathcal{A} \;:=\; \operatorname{span}\{a_1, \ldots, a_p\}.$$

For the random subspace signal $a(t)$, its covariance function is easily seen to be

$$r_a(t_1, t_2) \;:=\; \mathsf{E}[a(t_1)a(t_2)^*] \;=\; \sum_{k=1}^{p}\sum_{\ell=1}^{p} (R_u)_{k,\ell}\, a_k(t_1) a_\ell(t_2)^*,$$

where R_u is the covariance matrix of the random vector u. The covariance operator corresponding to the covariance function r_a is

$$(R_a x)(t) \;:=\; \int_0^T r_a(t,\tau) x(\tau)\, d\tau, \quad 0 \le t \le T.$$

A simple calculation shows that

$$R_a x \;=\; A R_u A^* x, \tag{3.2}$$

where the adjoint operator $A^*\colon L^2[0,T] \to \mathbb{C}^p$ is given by

$$A^* x \;=\; \begin{bmatrix} \langle x, a_1 \rangle \\ \vdots \\ \langle x, a_p \rangle \end{bmatrix},$$

and $\langle \cdot\,, \cdot \rangle$ denotes the standard inner product on $L^2[0,T]$,

$$\langle x, a \rangle \;:=\; \int_0^T x(t)a(t)^*\,dt.$$

We assume that R_u is invertible.

Suppose that the interference b is a second-order process with covariance function $r_b(t_1,t_2) := \mathsf{E}[b(t_1)b(t_2)^*]$ and corresponding covariance operator

$$(R_b x)(t) \;:=\; \int_0^T r_b(t,\tau)x(\tau)\,d\tau, \quad 0 \le t \le T.$$

It is easy to see that R_b is self adjoint. Next, if

$$\int_0^T \int_0^T |r_b(t,\tau)|^2\,dt\,d\tau \;<\; \infty,$$

then R_b is compact [2, pp. 86–87]. By the spectral theorem for compact, self-adjoint operators [2, p. 113], R_b has the representation

$$R_b x \;=\; \sum_{k=1}^{\infty} \beta_k \langle x, s_k \rangle s_k,$$

where the β_k are positive, nonincreasing eigenvalues with corresponding orthonormal eigenvectors s_k; i.e., $R_b s_k = \beta_k s_k$. Furthermore, we have the *orthogonal* direct sum decomposition [2, p. 115]

$$L^2[0,T] \;=\; \ker R_b \oplus \mathcal{B}, \tag{3.3}$$

where

$$\mathcal{B} \;:=\; \overline{\operatorname{span}\{s_k\}},$$

and the overbar denotes the closure.

If b is in fact zero-mean, mean-square continuous, then b has the Karhunen–Loève expansion

$$b(t) \;=\; \sum_{k=1}^{\infty} B_k s_k(t),$$

where

$$B_k \;=\; \int_0^T b(t)s_k(t)^*\,dt.$$

However, this is more than we need. All we really need is (3.3).

3.2. Identification of $P_{\mathcal{B}}^{\perp}(\mathcal{A})$

An important quantity in the design of matched subspace detectors is the subspace

$$\mathcal{G} \;:=\; P_{\mathcal{B}}^{\perp}(\mathcal{A}) \;=\; \operatorname{span}\{P_{\mathcal{B}}^{\perp}a_1, \ldots, P_{\mathcal{B}}^{\perp}a_p\}, \tag{3.4}$$

where $P_{\mathcal{B}}^{\perp}$ is the projection onto the orthogonal complement of $\mathcal{B}$. We assume $\mathcal{A} \cap \mathcal{B} = \{0\}$ (the zero subspace); this condition is necessary and sufficient to guarantee that the $P_{\mathcal{B}}^{\perp}a_k$ are linearly independent, and hence that $\dim \mathcal{G} = p$. How can we find $\mathcal{G}$ if we do not know $\mathcal{B}$?

Let us suppose that we can estimate the covariance function $r_y(t_1, t_2)$ from observed data. Assume that we know the a_k, σ^2, and that a, b, and n are uncorrelated. Then

$$R_y \;=\; R_a + R_b + \sigma^2 I.$$

Put

$$R \;:=\; R_a + R_b \;=\; R_y - \sigma^2 I.$$

Then we know R.

Theorem 3.1. *The operator $R\colon L^2[0,T] \to L^2[0,T]$ maps the subspace $\mathcal{G}$ one-to-one and onto the subspace $\mathcal{A}$. Furthermore, $\mathcal{G} = R^+(\mathcal{A})$, where R^+ denotes the pseudoinverse of R.*[4] *It then follows that*

$$\mathcal{G} \;=\; \operatorname{span}\{R^+a_1, \ldots, R^+a_p\}.$$

Remark. The point here is that by knowing the covariance function $r_y(t,\tau)$, the noise variance σ^2, and the basis waveforms a_k in (3.1), we can obtain a basis for $\mathcal{G}$. Having a basis for $\mathcal{G}$, we can project any waveform onto it by solving the appropriate matrix-column-vector normal equations. It is not necessary to know $\mathcal{B}$.

Proof of Theorem 3.1. We first show that for $g \in \mathcal{G}$, $Rg \in \mathcal{A}$. Since $Rg = R_a g + R_b g$, and since for all x, $R_a x \in \mathcal{A}$, it suffices to show that $R_b g = 0$. Now, $g \in \mathcal{G} \subset \mathcal{B}^\perp$. It follows from the orthogonal decomposition (3.3) that $\mathcal{B}^\perp = \ker R_b$; thus, for $g \in \mathcal{G}$, $R_b g = 0$. Hence, $Rg = R_a g \in \mathcal{A}$ as claimed.

Since $\dim \mathcal{G} = \dim \mathcal{A} < \infty$, if we can prove R is one-to-one on $\mathcal{G}$, then by [4, p. 81, Th. 9], it follows that R maps $\mathcal{G}$ onto $\mathcal{A}$. We proceed as follows to show R is one-to-one on $\mathcal{G}$. For $u, v \in \mathbb{C}^p$, put $g_1 = P_{\mathcal{B}}^\perp A u$ and $g_2 = P_{\mathcal{B}}^\perp A v$, and suppose $Rg_1 = Rg_2$. As just noted, this implies $R_a g_1 = R_a g_2$. Using (3.2), we have

$$AR_u(A^*P_{\mathcal{B}}^\perp A)u \;=\; AR_u(A^*P_{\mathcal{B}}^\perp A)v.$$

Since A is one-to-one and R_u is assumed invertible, we have

$$(A^*P_{\mathcal{B}}^\perp A)u \;=\; (A^*P_{\mathcal{B}}^\perp A)v.$$

Since $A^*P_{\mathcal{B}}^\perp A = (P_{\mathcal{B}}^\perp A)^*(P_{\mathcal{B}}^\perp A)$, and since the $P_{\mathcal{B}}^\perp a_k$ are linearly independent, $P_{\mathcal{B}}^\perp A$ is one-to-one, and then so is[5] $(P_{\mathcal{B}}^\perp A)^*(P_{\mathcal{B}}^\perp A)$. Thus, $u = v$ and then $g_1 = g_2$.

Since R maps $\mathcal{G}$ one-to-one and onto $\mathcal{A}$, there is an inverse map from $\mathcal{A}$ onto $\mathcal{G}$; i.e., for every $g \in \mathcal{G}$ there is exactly one $a \in \mathcal{A}$ such that $Rg = a$. If $g \in (\ker R)^\perp$, then $g = R^+a$. We now show that if $g = P_{\mathcal{B}}^\perp A u$ for some u, then g is orthogonal to $\ker R$.

[4] For a bounded operator R, the domain of R^+ is the set of all $y \in L^2[0,T]$ such that the projection of y onto the range of R exists. Denoting this projection by $\widehat{y}$, R^+y is defined to be the minimum-norm solution of $Rx = \widehat{y}$. By definition of $\widehat{y}$, it is in the range of R, and so there is at least one solution x_0 such that $Rx_0 = \widehat{y}$. Since R is a bounded operator, $\ker R$ is closed. Since $L^2[0,T]$ is complete, the projection theorem [5, §3.3] shows that $L^2[0,T] = \ker R \oplus (\ker R)^\perp$. It is then easily seen that the projection of x_0 onto $(\ker R)^\perp$ is the unique, minimum-norm solution. We shall be concerned with the case in which y is already in the range of R so that $\widehat{y} = y$. For such y, R^+y is defined.

[5] Use the easily verified fact that for any operator D with adjoint D^*, $\ker D = \ker D^*D$.

Fix any $x \in \ker R$ so that $Rx = 0$. Then $R_a x = -R_b x$ is an element of $\mathcal{A} \cap \mathcal{B} = \{0\}$. Thus, $R_a x = R_b x = 0$. In particular, $R_b x = 0$ implies $x \in \mathcal{B}^\perp$. Now write

$$\langle g, x \rangle \;=\; \langle P_{\mathcal{B}}^\perp A u, x \rangle \;=\; \langle A u, P_{\mathcal{B}}^\perp x \rangle \;=\; \langle A u, x \rangle \;=\; \langle u, A^* x \rangle,$$

which equals zero because $0 = R_a x = A R_u A^* x$ implies $A^* x = 0$ on account of the linear independence of the a_k and the invertibility of R_u. Thus, $\mathcal{G} \subset (\ker R)^\perp$. □

We now turn to the problem of finding R_u.

Theorem 3.2. *The covariance matrix R_u can be obtained via*

$$R_u \;=\; (A^* R^+ A)^{-1}.$$

Proof. It suffices to show that $A^* R^+ A R_u = I$, which we do by showing that $A^* R^+ A R_u v = v$ for arbitrary $v \in \mathbb{C}^p$. Given $v \in \mathbb{C}^p$, $A R_u v \in \mathcal{A}$. By Theorem 3.1, $R^+ A R_u v \in \mathcal{G}$. Hence,

$$R^+ A R_u v \;=\; P_{\mathcal{B}}^\perp A u, \quad \text{for some } u \in \mathbb{C}^p. \tag{3.5}$$

From the definition of pseudoinverse, $R P_{\mathcal{B}}^\perp A u$ is equal to the projection of $A R_u v$ onto the range of R. Since $A R_u v \in \mathcal{A}$, which, by Theorem 3.1, is a subset of the range of R, $R P_{\mathcal{B}}^\perp A u = A R_u v$. Now write

$$\begin{aligned} A R_u v &= R P_{\mathcal{B}}^\perp A u \\ &= R_a P_{\mathcal{B}}^\perp A u \\ &= A R_u A^* P_{\mathcal{B}}^\perp A u. \end{aligned}$$

Since A is one-to-one, and since R_u is assumed invertible, $v = A^* P_{\mathcal{B}}^\perp A u$. Using (3.5), write

$$A^* R^+ A R_u v \;=\; A^* P_{\mathcal{B}}^\perp A u \;=\; v.$$ □

4. Waveform Sets with Maximum Mutual Information

Consider the received waveform

$$y(t) \;=\; (Au)(t) + n(t),$$

where

$$(Au)(t) \;:=\; \sum_{k=1}^{p} u_k a_k(t). \tag{4.1}$$

We assume that the waveforms a_k are linearly independent and that $u := [u_1, \ldots, u_p]'$ is $N(0, R_{uu})$ (R_{uu} positive definite) and independent of the zero-mean, white Gaussian noise $n(t)$ with power spectral density $S_n(f) \equiv \sigma^2$. Our goal is to choose the waveforms a_k so as to maximize the average mutual information between u and y, denoted by $I(u \wedge y)$. The idea is to choose waveforms a_k so that the received waveform y provides as much information as possible about the weights u_k.

Direct evaluation of $I(u \wedge y)$ is complicated by the fact that u is a column vector and y is a continuous-time random process. In Section 4.1, we show that $I(u \wedge y) = I(u \wedge v)$ where v is a column vector related to y. It will be seen that u and v are jointly Gaussian. Hence, the joint distribution of u and v is completely determined by the covariance matrices R_{uu} and R_{vv} and the cross-covariance matrix R_{uv}. In Section 4.2, R_{uv} and R_{vv} are obtained. Then $I(u \wedge v)$ is expressed in terms of known quantities in Section 4.3. Finally, in Section 4.4, the optimization problem is formulated and solved in closed form.

4.1. Reduction to a Finite-Dimensional Problem

Let $\mathcal{A} := \mathrm{span}\{a_1, \ldots, a_p\}$, and let $P_{\mathcal{A}}$ denote the projection operator onto $\mathcal{A}$. Since $y = Au + n$, where $Au \in \mathcal{A}$, we have

$$\widehat{y} \;:=\; P_{\mathcal{A}} y \;=\; Au + P_{\mathcal{A}} n$$

and

$$\widetilde{y} \;:=\; y - \widehat{y} \;=\; n - P_{\mathcal{A}} n \;=\; P_{\mathcal{A}}^{\perp} n,$$

where $P_{\mathcal{A}}^{\perp}$ denotes the projection onto $\mathcal{A}^{\perp}$, the orthogonal complement of $\mathcal{A}$. Since $\widehat{y}$ and $\widetilde{y}$ are uncorrelated and jointly Gaussian, they are independent.

Using the fact that the mapping $y \mapsto (\widehat{y}, \widetilde{y}\,)$ is invertible, along with a standard identity, we have

$$I(u \wedge y) \;=\; I(u \wedge (\widehat{y}, \widetilde{y}\,)) \;=\; I(u \wedge \widehat{y}\,) + I(u \wedge \widetilde{y}\,|\widehat{y}\,).$$

To see that this last term is zero, observe that

$$\begin{aligned} I(u \wedge \widetilde{y}\,|\widehat{y}\,) &= H(\widetilde{y}\,|\widehat{y}\,) - H(\widetilde{y}\,|\widehat{y}, u) \\ &= H(\widetilde{y}\,) - H(\widetilde{y}\,), \quad \text{by independence,} \\ &= 0. \end{aligned}$$

Thus,

$$I(u \wedge y) \;=\; I(u \wedge \widehat{y}\,). \tag{4.2}$$

We next show that the waveform $\widehat{y}$ is equivalent to the column vector

$$A^*y \;=\; \begin{bmatrix} \langle y, a_1 \rangle \\ \vdots \\ \langle y, a_p \rangle \end{bmatrix},$$

where $\langle \cdot\,, \cdot \rangle$ denotes the inner product,

$$\langle y, a \rangle \;:=\; \int y(t)a(t)^*\,dt.$$

By equivalent we mean that $\widehat{y}$ can be obtained as a function of A^*y and A^*y can be obtained as a function of $\widehat{y}$. To see this, first recall that $P_{\mathcal{A}} = A(A^*A)^{-1}A^*$ [5, pp. 160–161]. Thus, $\widehat{y} = P_{\mathcal{A}}y$ is a function of A^*y. Conversely,

$$A^*\widehat{y} \;=\; A^*A(A^*A)^{-1}A^*y \;=\; A^*y.$$

Since $\widehat{y}$ and A^*y are equivalent,

$$I(u \wedge \widehat{y}\,) \;=\; I(u \wedge A^*y). \tag{4.3}$$

To conclude this section, we define v in terms of A^*y by the invertible transformation

$$v \;:=\; (A^*A)^{-1/2}A^*y.$$

Hence, $I(u \wedge A^*y) = I(u \wedge v)$. Putting this together with (4.2) and (4.3), we have $I(u \wedge y) = I(u \wedge v)$ as required.

4.2. The Joint Distribution of u and v

Observe that

$$v \;:=\; (A^*A)^{-1/2}A^*y \;=\; (A^*A)^{1/2}u + w,$$

where $w := (A^*A)^{-1/2}A^*n$ is a p-dimensional $N(0, \sigma^2 I)$ random vector. In now follows that u and v are jointly Gaussian with

$$R_{uv} \;=\; R_{uu}(A^*A)^{1/2} \tag{4.4}$$

and

$$\begin{aligned} R_{vv} &= (A^*A)^{1/2}R_{uu}(A^*A)^{1/2} + \sigma^2 I \\ &= (A^*A)^{1/2}R_{uu}^{1/2}[I + S^{-1}]R_{uu}^{1/2}(A^*A)^{1/2}, \end{aligned} \tag{4.5}$$

where S is the **SNR matrix**,

$$S \;:=\; \frac{R_{uu}^{1/2}(A^*A)R_{uu}^{1/2}}{\sigma^2}. \tag{4.6}$$

4.3. Evaluation of the Average Mutual Information

An easy calculation, e.g., [11, Section V], shows that the average mutual information between u and v is

$$I(u \wedge v) \;=\; \frac{1}{2}[\ln\det R_{vv} - \ln\det Q_{vv}],$$

where [11, eq. (20)]

$$\begin{aligned} Q_{vv} \;&:=\; R_{vv} - R'_{uv}R_{uu}^{-1}R_{uv} \\ &=\; R_{vv}^{1/2}(I - C'C)R_{vv}^{1/2}, \end{aligned}$$

and

$$C \;:=\; R_{uu}^{-1/2}R_{uv}R_{vv}^{-1/2} \tag{4.7}$$

is the **coherence matrix**. It follows that

$$I(u \wedge v) \;=\; -\tfrac{1}{2}\ln\det(I - C'C) \;=\; -\tfrac{1}{2}\ln\det(I - CC').$$

Using (4.7) along with (4.4) and (4.5), it is easy to check that $CC' = (I + S^{-1})^{-1}$. Using the matrix inversion formula [8],

$$I - CC' \;=\; I - [I - (I + S)^{-1}] \;=\; (I + S)^{-1}.$$

Thus,

$$I(u \wedge v) \;=\; \tfrac{1}{2}\ln\det(I + S).$$

4.4. The Optimization Problem

It is instructive to consider the case in which $p = 1$ in (4.1). Then the SNR matrix S in (4.6) is the scalar SNR,

$$S \;=\; \frac{R_{uu}\|a_1\|^2}{\sigma^2}.$$

It is then clear that $I(u \wedge v)$ is monotonic increasing with $A^*A = \|a_1\|^2$. Hence, in order to have $\max_A I(u \wedge v)$ finite, we must impose some kind of constraint on the energy of the signaling waveforms.

Let

$$L \;:=\; R_{uu}^{1/2}(A^*A)R_{uu}^{1/2}$$

denote the numerator in (4.6) so that $S = L/\sigma^2$. Consider the problem

$$\max_A \tfrac{1}{2}\ln\det(I + L/\sigma^2) \quad \text{subject to } \operatorname{tr} L \le \mathcal{E}, \tag{4.8}$$

where $\mathcal{E}$ is a constraint on the allowable energy of the signaling waveforms. Now observe that for any orthogonal matrix P ($P'P = PP' = I$), (4.8) is equivalent to

$$\max_A \tfrac{1}{2}\ln\det(I + P'LP/\sigma^2) \quad \text{subject to } \operatorname{tr}(P'LP) \le \mathcal{E}.$$

Hence, if P is chosen to diagonalize L, then

$$P'LP \;=\; \Lambda \;=\; \operatorname{diag}(\lambda_1, \ldots, \lambda_p),$$

and (4.8) becomes

$$\max_A \frac{1}{2}\sum_{k=1}^{p} \ln(1+\lambda_k/\sigma^2) \quad \text{subject to } \sum_{k=1}^{p} \lambda_k \le \mathcal{E},$$

where the λ_k are the eigenvalues of L. Using Lagrange multipliers, it is an easy exercise to show that the optimal choices for the λ_k are $\lambda_k = \mathcal{E}/p$ for all k and that $I(u\wedge v) = (p/2)\ln(1+\mathcal{E}/p\sigma^2)$. It is easy to see that if $A^*A = (\mathcal{E}/p)R_{uu}^{-1}$, then $L = R_{uu}^{1/2}(A^*A)R_{uu}^{1/2} = (\mathcal{E}/p)I$ is diagonal with all eigenvalues equal to $\mathcal{E}/p$. To conclude, we observe that if $\tilde{a}_1,\ldots,\tilde{a}_p$ is any orthonormal basis of waveforms with corresponding operator $\tilde{A}$ given as in (4.1), then $A := \sqrt{\mathcal{E}/p}\,\tilde{A}R_{uu}^{-1/2}$ solves the problem. In other words, we should take

$$a_k(t) \;=\; \sqrt{\frac{\mathcal{E}}{p}}\sum_{i=1}^{p}(R_{uu}^{-1/2})_{ik}\,\tilde{a}_i(t).$$

Remark. In general, for any operator $\check{A}$ defined similarly to (4.1), if we take $A := \check{A}R_{uu}^{-1/2}$, then

$$y \;=\; Au+n \;=\; \check{A}R_{uu}^{-1/2}u+n.$$

If we put $\check{u} := R_{uu}^{-1/2}u$, then

$$y \;=\; \check{A}\check{u}+n,$$

where the covariance matrix of $\check{u}$ is I. In other words, we are whitening the input.

4.5. Discussion of the Constraint in (4.8)

It is rather obvious that we used the constraint

$$\operatorname{tr} L \;=\; \operatorname{tr}(R_{uu}^{1/2}(A^*A)R_{uu}^{1/2}) \;\le\; \mathcal{E}$$

to make (4.8) easy to solve in closed form. Here is a different approach to the problem that *appears* to use the more natural constraint

$$\operatorname{tr}(A^*A) \;=\; \sum_{k=1}^{p}\|a_k\|^2 \;\le\; \mathcal{E}.$$

Rewrite $y = Au+n$ as

$$y \;=\; AR_{uu}^{1/2}R_{uu}^{-1/2}u+n.$$

If we put

$$\check{A} \;:=\; AR_{uu}^{1/2} \tag{4.9}$$

and $\check{u} := R_{uu}^{-1/2}u$, then we have the model $y = \check{A}\check{u}+n$, where now $\check{u}$ has covariance matrix I. Since u and $\check{u}$ are related by an invertible transformation, $I(u\wedge y) = I(\check{u}\wedge y)$. We would then say that there is no loss of generality in taking $R_{uu} = I$ and proceed as above and impose the apparently natural constraint $\operatorname{tr}(A^*A)\le\mathcal{E}$. However, this is misleading because the foregoing A is actually $\check{A}$ in (4.9). Using (4.9), $\operatorname{tr}(\check{A}^*\check{A}) = \operatorname{tr}(R_{uu}^{1/2}(A^*A)R_{uu}^{1/2})$.

4.6. Future Work

One obvious extension is to replace the constraint in (4.8) with the more natural constraint

$$\mathrm{tr}(A^*A) \;=\; \sum_{k=1}^{p} \|a_k\|^2 \;\le\; \mathcal{E}.$$

However, this seems rather challenging.

In a multipath channel, the received waveform would again be $y = Au + n$, but now the a_k would have the form $a_k(t) = a(t-(k-1)T)$, $k = 1, \ldots, p$, where T is proportional to the reciprocal of the bandwidth of the basic waveform $a(t)$, and p is proportional to the product of the bandwidth and the channel multipath spread as in Section 1. In this case, the foregoing analysis does not immediately apply since there is now the additional constraint that the a_k be shifts of a basic pulse a.

5. Fusion of Decentralized Decisions

5.1. A Preliminary: Detection with an Arbitrary Discrete Random Variable

Let Z be a discrete random variable taking N distinct values with positive probability under each of two hypotheses. Let $\mathcal{Z}$ denote the set of N values taken by Z, and let

$$L(z) \;:=\; \frac{\wp_1(Z=z)}{\wp_0(Z=z)}, \quad z \in \mathcal{Z}, \tag{5.1}$$

denote the likelihood ratio of Z. Without loss of generality, we enumerate the elements of $\mathcal{Z}$, say $z_1, \ldots, z_N$, so that

$$L(z_1) \;\le\; \cdots \;\le\; L(z_N).$$

If $\eta = L(z_k) > L(z_{k-1})$, then the probability of false alarm is

$$p_{\rm FA}(\eta) \;=\; \wp_0(L(Z) \ge \eta) \;=\; \sum_{i=k}^{N} \wp_0(Z = z_i) \tag{5.2}$$

and

$$p_{\rm D}(\eta) \;=\; \wp_1(L(Z) \ge \eta) \;=\; \sum_{i=k}^{N} \wp_1(Z = z_i).$$

Note that $p_{\rm FA}$ and $p_{\rm D}$ are nonincreasing, left-continuous functions of η. In fact, these functions are piecewise constant with jumps at the values $\eta = L(z_k)$.

5.2. Fusion of Local Binary Decisions

Consider a collection of n decentralized sensors, each making its own binary decision $D_i = 0$ or 1 about whether the underlying hypothesis is 0 or 1. At a fusion center, the decentralized decisions are collected into the discrete random variable $Z := [D_1, \ldots, D_n]'$ taking $N = 2^n$ distinct values. The centralized decision of the fusion center may then be treated as in Section 5.1.

5.3. Quantization for ROC Approximation

Let Y be a random vector taking values in $\mathbb{R}^d$, and let $A_1, \ldots, A_N$ be a partition of $\mathbb{R}^d$. Put $Z = z_k$ if $Y \in A_k$ so that Z is a discrete random variable as above. If the sets A_k satisfy $\wp_0(Y \in A_k) = \Delta\alpha$ for some small $\Delta\alpha$, then the ROC curve, which plots $p_{\rm D}(\eta)$ versus $p_{\rm FA}(\eta)$, will have points closely spaced on the horizontal axis.

As an example, consider a real-valued random variable Y with cumulative distribution function $F_i(y) = \wp_i(Y \le y)$, $i = 0, 1$. Put $\Delta\alpha = 1/N$, and for $k = 1, \ldots, N-1$, let y_k solve $F_0(y_k) = k\Delta\alpha$. Put $A_k := (y_{k-1}, y_k]$, where $y_0 := -\infty$, and $A_N = (y_{N-1}, \infty)$. Then $\wp_0(Z = z_k) = \Delta\alpha$. If (5.1) does not hold, it may be necessary to renumber the z_k. In any case, the distinct values of $p_{\rm FA}(\eta)$ in (5.2) will be spaced $\Delta\alpha$ apart.

5.4. Quantization for Likelihood Ratio Approximation

Let Y_j be a continuous-valued measurement available at sensor j, $j = 1, \ldots, n$. Let $L(y_1, \ldots, y_n)$ denote the likelihood ratio based on the joint distribution of the Y_j. Assume each sensor is equipped with a fine partition of intervals. If sensor j observes $Y_j \in (a_{j,k_j}, b_{j,k_j}]$,

it transmits only the index k_j. Since the fusion center cannot evaluate $L(Y_1,\ldots,Y_n)$, it uses the value

$$L\Big(\frac{a_{1,k_1}+b_{1,k_1}}{2},\ldots,\frac{a_{n,k_n}+b_{n,k_n}}{2}\Big).$$

5.5. Many Independent Sensors

Let A_k be such that

$$\alpha_k \;:=\; \wp_0(Y_k\in A_k) \;<\; \wp_1(Y_k\in A_k) \;:=\; \beta_k.$$

Such a test is said to be **unbiased**. Put

$$X_n \;:=\; \frac{1}{n}\sum_{k=1}^{n} I_{A_k}(Y_k),$$

and observe that

$$\mathsf{E}_0[X_n] \;=\; \frac{1}{n}\sum_{k=1}^{n}\alpha_k \;<\; \frac{1}{n}\sum_{k=1}^{n}\beta_k \;=\; \mathsf{E}_1[X_n].$$

If for some $\varepsilon>0$ and some η, $\mathsf{E}_0[X_n]<\eta-\varepsilon$ and $\eta+\varepsilon<\mathsf{E}_1[X_n]$ for all n, then under reasonable conditions a suitable weak law of large numbers will hold so that

$$\wp_0(X_n>\eta)\to 0 \quad\text{and}\quad \wp_1(X_n>\eta)\to 1.$$

In other words, asymptotically, as long as each sensor doesn't do too badly, the fusion center can obtain arbitrarily small probability of false alarm and arbitrarily high probability of detection. Of course, for finite n, we want to make $\mathsf{E}_0[X_n]$ and $\mathsf{E}_1[X_n]$ far apart. For example,

$$\mathsf{E}_1[X_n]-\mathsf{E}_0[X_n] \;=\; \frac{1}{n}\sum_{k=1}^{n}(\beta_k-\alpha_k).$$

If each α_k is fixed, then each β_k can be maximized by choosing A_k according to the Neyman–Pearson Lemma.

References

[1] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

[2] I. Gohberg and S. Goldberg, *Basic Operator Theory*. Boston: Birkhäuser, 1980.

[3] J. A. Gubner and L. L. Scharf "Detection of constrained subspace signals in additive infinite-dimensional interference and noise," *IEEE Trans. Inform. Theory*, in review.

[4] K. Hoffman and R. Kunze, *Linear Algebra*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1971.

[5] D. G. Luenberger, *Optimization by Vector Space Methods*. New York: Wiley, 1969.

[6] B. Picinbono, *Random Signals and Systems*. Englewood Cliffs, NJ: Prentice Hall, 1993.

[7] A. M. Sayeed and B. Aazhang, "Joint multipath-Doppler diversity in mobile wireless communications," *IEEE Trans. Commun.*, vol. 47, no. 1, pp. 123–132, Jan. 1999.

[8] L. L. Scharf, *Statistical Signal Processing*. Reading, MA: Addison-Wesley, 1991.

[9] L. L. Scharf and B. Friedlander, "Matched subspace detectors," *IEEE Trans. Signal Processing*, vol. 42, no. 8, pp. 2146–2157, Aug. 1994; see also "Corrections to 'Matched subspace detectors,'" vol. 45, no. 6, p. 1669, June 1997.

[10] L. L. Scharf and M. L. McCloud, "Blind adaptation of zero forcing projections and oblique pseudo-inverses for subspace dtection and estimation when interference dominates noise," *IEEE Trans. Signal Processing*, vol. 50, no. 12, pp. 2938–2946, Dec. 2002.

[11] L. L. Scharf and C. T. Mullis, "Canonical coordinates and the geometry of inference, rate, and capacity," *IEEE Trans. Signal Processing*, vol. 48, no. 3, pp. 824–831, Mar. 2000.

[12] X. Wang and H. V. Poor, "Blind multiuser detection: A subspace approach," *IEEE Trans. Inform. Theory*, vol. 42, no. 1, pp. 677–690, Mar. 1998.

ISP for Communication Links

Mathematical Modeling, Problem Formulation, and Analysis

Chad M. Spooner
Mission Research Corporation

May 19, 2003

Version 1.0

Abstract

The concept of integrating sensing with processing for general communication problems is developed. The fundamental notion of varying basic system parameters—including modulation type, carrier frequency, bit/symbol rate, coding scheme, and so on—in a manner that is dependent on a regular sensing of the operating environment is introduced. This notion is abstracted mathematically, resulting in a concrete model for a time-varying communication system in the presence of a time-varying channel. The goal of the research is to find classes of high-capacity time-varying systems that can react favorably to time-varying channels. Using the developed mathematical abstraction as a framework and this overarching goal as a guiding principle, a sequence of mathematical problems is posed. The practicality of real time-varying systems that are based on the abstraction is discussed in light of modern communication system technologies such as high-speed DSP and software radio.

Contents

List of Figures

List of Tables

1 Introduction

This document is a record of MRC's ISP program effort that relates to communication systems. In particular, a specific notion of the integration of sensing and processing (ISP) for arbitrary communication links is proposed. This notion is set against the backdrop of traditional communication-system design philosophy and is used to develop mathematical abstractions that lead to specific mathematical problems of ISP interest. The posing and study of these problems makes up the ISP work related to communication systems.

The remainder of this document is organized as follows. The traditional approaches to communication-system design are briefly described in Section 2 and ISP approaches are defined in Section 3. The ISP approaches are used to develop overarching mathematical abstractions of communication systems under ISP in Section 4, and specific mathematical problems are then posed in Section 5. Section 6 addresses the gap between the mathematical problems and the engineering design of real systems that are capable of achieving some or all of the ISP gains in throughput, robustness to channel impairments, or error probability. Section 7 provides mathematical problem analysis, Section 8 contains a discussion of results and some numerical examples, and concluding remarks are provided in Section 9. Appendix A contains some relevant information-theoretic definitions and results.

2 Conventional Communication System Design Philosophy

Conventional design approaches can be divided into the traditional approach and the modern approaches.

2.1 Traditional Approach

In the traditional approach to communication-system design, modeling and measurement are used to characterize (perhaps incompletely) the physical channel and the source. Then each block in the basic block diagram in Figure 1 is separately designed and optimized. A basic difficulty with this approach is that it cannot adapt to deviations from the assumed channel model. That is, it can neither take advantage of unusually good conditions nor defend against bad conditions that render the nominal system completely ineffective (high error rate).

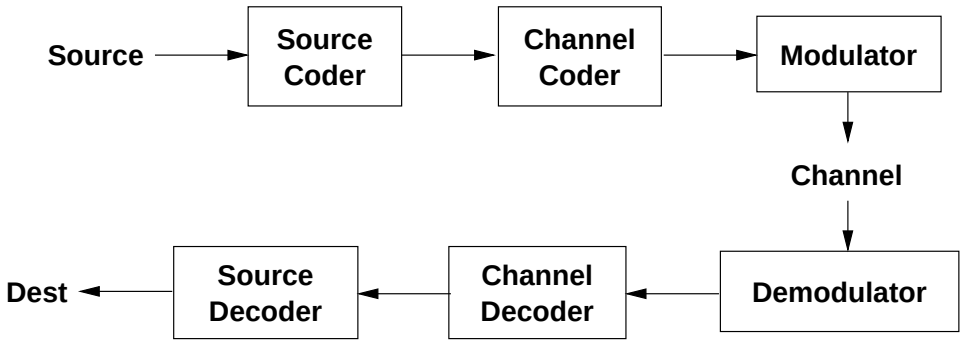


Figure 1: Basic communication-system block diagram.

2.2 Modern Approaches

In more recent approaches [17]–[25], the traditional approach has been modified by allowing one or more of the basic blocks in Figure 1 to possess an adaptive character or by combining two or more of the blocks and treating the result as an integral subsystem to be optimized. Adaptive equalizers are examples of the former and the most visible example of the latter is trellis-coded modulation, arising from the combination of the channel coder and the modulator, as shown in Figure 2.

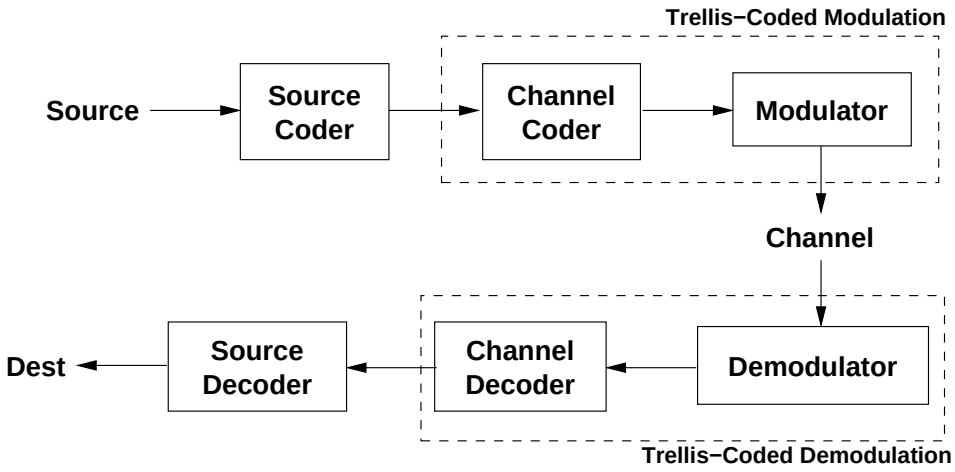


Figure 2: Block diagram for a communication system employing trellis-coded modulation.

Another important example of the more flexible modern approach is multicarrier modulation or OFDM. The signal can be viewed as a set of adjacent-channel digital QAM signals with identical symbol rates. The constellation used in each subcarrier can be varied to adapt to changing channel conditions, resulting in variable bit rate across the subchannels.

Although the modern approaches do provide means for adapting to changing channels (and possibly to changing data-rate and quality-of-service demands), they do not go as far as possible in integrating environment sensing with processing. In particular, many other parameters may be adjusted to react to time-varying channels, including modulation type (not just the constellation), coding schemes and parameters, center frequency, antenna pattern, RF bandwidth, spreading gain, etc.

3 Integrating Sensing and Processing for Communication Systems

To integrate sensing with processing in the communication-system context, two things are required: (1) the ability to sense one or more aspects of the environment, and (2) the ability to alter one or more aspects of the processing in response to sensing. For our purposes, the environment is not limited to the physical channel, but also includes possibly time-varying user demands on data rate and error performance.

We can divide the class of ISP-enabled communication systems into those that require significant cooperation between source and destination and those that do not. The former will be called

cooperative ISP systems (CISPS), and the latter *autonomous ISP systems* AISPS. In autonomous systems, the destination senses the environment on its own and makes its own choices for operating parameters accordingly. No feedback between destination and source is required, as illustrated in Figure 3. Of course, this severely limits the set of adjustable parameters.

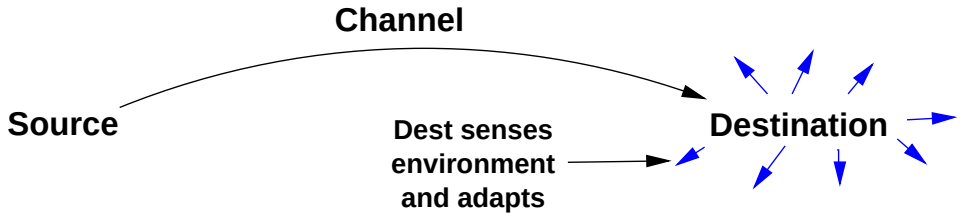


Figure 3: Illustration of the autonomous ISP communication system concept.

In cooperative systems, the destination senses the environment with cooperation from the source and makes choices that are communicated to the source or that must be negotiated with the source. This requires either a logical or physical subchannel for passing side information back and forth, as illustrated in Figure 4.

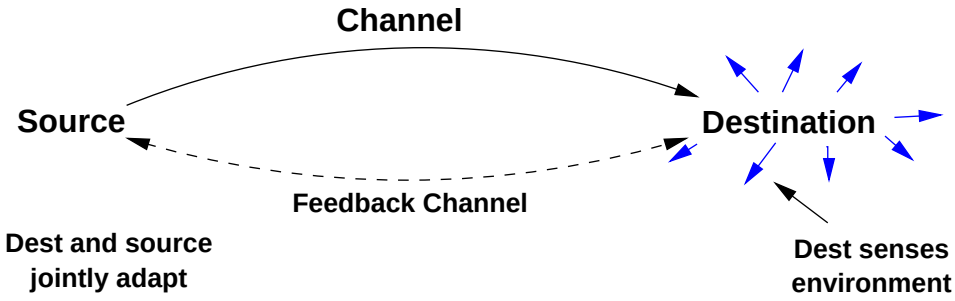


Figure 4: Illustration of the cooperative ISP communication system concept.

For all the possible ISP-enabled communication systems, we propose the following concept of operation:

> *Let the set of adjustable parameters at the destination be denoted by P_a, and let the destination perform a set of measurements each T seconds. Based on the measurements, adjust one or more parameters in P_a to improve error performance and/or increase data rate.*

A major goal of this work is the development of a suitable mathematical framework for predicting performance improvements due to the use of the proposed ISP concept of operations.

3.1 Conceptual ISP Examples

In this section we provide a few examples of the ISP system concepts to fix the autonomous and cooperative notions.

Simple Autonomous Parameter Adjustment.
In this first example, the destination senses the environment and makes decisions regarding the

best parameter adjustment but does not communicate the adjustment to the source. The destination senses the channel by adjusting its position in space and measuring the quality of the communication for the new position. Since each new position induces a new channel, one of the new positions will correspond to the best new performance level. The sensor then moves to this position.

To be more specific, let the sensor position at time t be denoted by $\boldsymbol{p}(t)$ and the number of candidate positions be restricted to P_n. Let the P_n vectors $\boldsymbol{r}_j$, $j = 1, 2, \ldots, n$ represent points on a sphere centered at the origin and with radius r meters. Then the sensor measures communication quality at each position $\boldsymbol{p}(t) + \boldsymbol{r}_j$ and selects the position with the highest quality for $\boldsymbol{p}(t + T)$.

Note that the radius r need not be particularly large for this scheme to work well. For instance, if the channel experiences deep, long fades, small changes in position can move the sensor completely out of the fade, increasing SNR by tens of dB. Note also that the 'autonomous' label for this kind of ISP does not preclude the use of repeated known training sequences sent by the source. Channel or BER estimates estimates based on such training sequences represent two quality measures. A measure that is independent of pilot symbols or training sequences is the constellation-quality measure that assesses the tightness of the constellation cluster elements at the destination.

Random Cooperative Parameter Adjustment.

In this simplest CISPS, the destination assesses the communication quality through some means (use of periodically repeated pilot symbols or training sequences or constellation quality measures), and when the quality is judged poor, the destination chooses a set of system parameters at random and relays this information to the source through the feedback channel. A limiting case corresponds to continuous random parameter adjustment in which the quality is always judged to be poor.

The amount of information to be sent to the source through the feedback channel can actually be quite small if the source and destination each have access to a table of system-parameter options. The random destination decision could then be communicated to the source by the transmission of a single small-integer table index. To be specific, the CISPS could use one of four parameter sets shown in Table 1. Note that these parameter sets include differences in modulation type, coding scheme, center frequency, and symbol rate (bandwidth). The 2FSK parameter set is more likely to result in acceptable communication for the poorer channels, whereas the 64QAM parameter set will maximize data rate for the better channels.

Preset Cooperative Parameter Adjustment.

In the next CISPS, the destination again measures the communication quality and makes a decision regarding whether to switch parameter sets. Here, however, the destination can choose only two new parameter sets for each current parameter set. One of the sets corresponds to degraded quality and the other to improved quality relative to the previous quality assessment. Thus, the parameter-set index is a Markov process with transition probabilities jointly specified by the stochastic channel model and application design constraints, as illustrated in Figure 5 for the parameter sets in Table 1. The basic idea is that the system senses the environment and tries to match its parameters to the environment but only has a limited number of choices and cannot select the parameters arbitrarily, but must walk up (or down) the sequence of progressively more (or less) capable system parameter sets.

Optimal Cooperative Parameter Adjustment.

In the third CISPS, the destination attempts to select the best parameter set from its allowable sets.

Set No.	Modulation Type	Coding Scheme	Carrier Frequency	Symbol Rate
1	2FSK	Conv., $R = 1/2$	200 MHz	20 kHz
2	OOK	None	250 MHz	30 kHz
3	QPSK	Conv., $R = 4/5$	300 MHz	50 kHz
4	64QAM	Conv., $R = 4/5$	300 MHz	50 kHz

Table 1: Four example parameter sets for the random cooperative parameter-adjustment example.

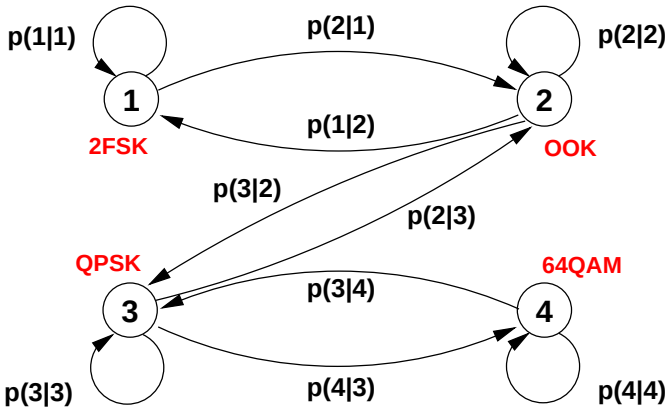


Figure 5: Illustration of parameter-set Markov process.

Thus, all transitions between parameter-set states are potentially valid. The probability $p(j|k)$ of transitioning from state k to state j is simply the probability of state j, $p(j)$. This scheme may be preferable when the channel is sensed with very little error, but when the sensing is error-prone or crude, the previous CISPS may be superior because the system is prohibited from making unwise radical changes in the parameter set.

3.2 Summary of Basic ISP Approach

The basic ISP notion advanced in this report is the periodic sensing of a time-varying environment combined with a means for modifying communication-link processing and transmission parameters accordingly. The concept generalizes modern communication-system notions such as rate-adaptation and multi-carrier modulation by allowing virtually any parameter to be modified, such as carrier frequency, modulation type (not limited to constellation type), coding scheme and parameters, etc. For example, a system may adaptively switch between frequency hopping, direct-sequence spread spectrum, and large-alphabet digital QAM in response to the presence of narrowband interferers, shadowing or fading, high SNR, or time-varying throughput and error constraints.

4 Mathematical Abstraction

In this section we set out to abstract our ISP communication problem in general mathematical terms. This abstraction will form the basis for posing interesting and relevant mathematical prob-

lems.

First, note that each system parameter set effectively induces a discrete channel, as illustrated in Figure 6. Often we will be interested in the subclass of *discrete memoryless channels* (DMCs), for which each output is statistically dependent only on the corresponding input and not on previous inputs or outputs. A general DMC is characterized by the *input alphabet* $x_1, x_2, \ldots, x_K$, the *output alphabet* $y_1, y_2, \ldots, y_J$, the *channel transition probabilities* $p_{j|k}$,

$$p_{j|k} \triangleq \text{Prob}(Y = y_j | X = x_k),$$

and the *prior probabilities* $q_1, q_2, \ldots, q_K$, as illustrated by the graph in Figure 7. The probability $p_{j|k}$ is the probability that the channel output is decided as y_j given that the channel input was x_k. The probability q_i is the prior probability of transmitting the letter x_i.

The transition probabilities are determined by the physical channel, the modulation, and the demodulator. For example, consider uncoded BPSK signaling on the additive white Gaussian noise (AWGN) channel with perfectly coherent demodulation and equiprobable inputs. Then $J = K = 2$ and the probability of bit error is given by the well-known formula

$$P_{e,BPSK} \approx Q\left(\sqrt{\frac{2E_b}{N_0}}\right),$$

where E_b is the energy per bit, N_0 is the noise spectral density, and

$$Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-u^2}\, du.$$

Denoting x_1 as 0, x_2 as 1, y_1 as 0, and y_2 as 1, we obtain the binary DMC shown in Figure 8.

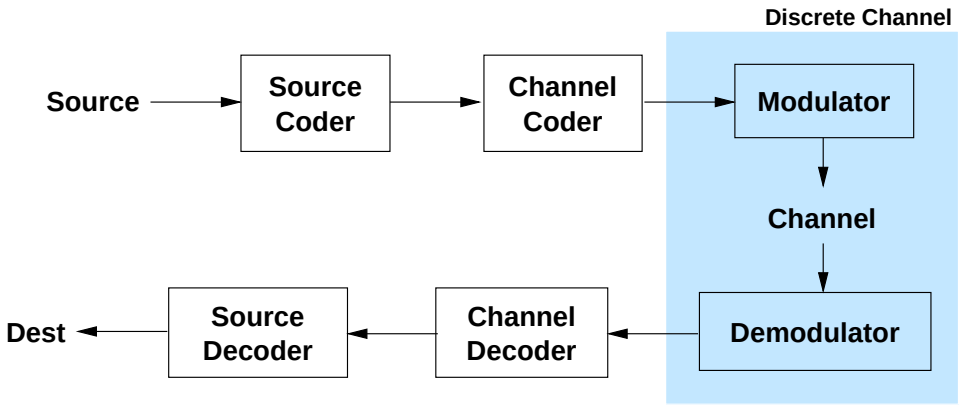


Figure 6: The basic communication system model with induced DMC highlighted.

Time-Varying Systems.

For our time-varying communication-system setup, the induced DMC is a function of time. That is, the alphabets, alphabet sizes, and transition probabilities are functions of time. To model the interval-oriented aspect of our particular situation, let t index each interval of length T seconds. The the DMC is considered constant on each interval t. For the tth interval, the input alphabet size

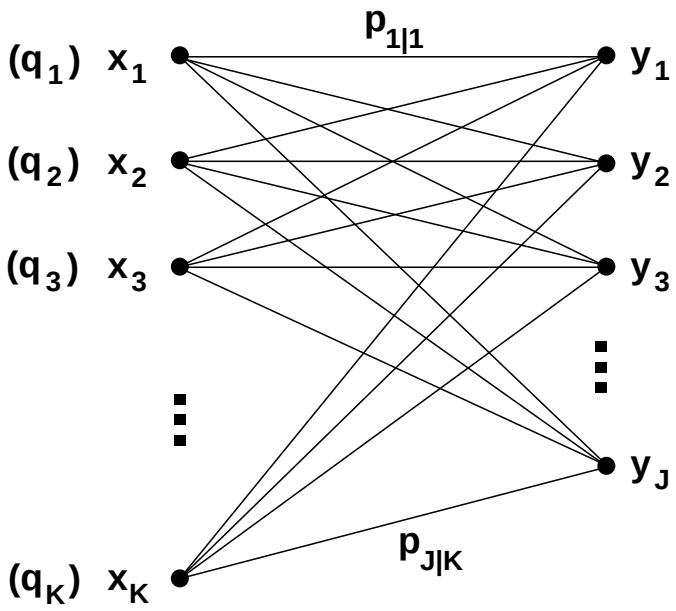


Figure 7: A diagram of a generic discrete memoryless channel.

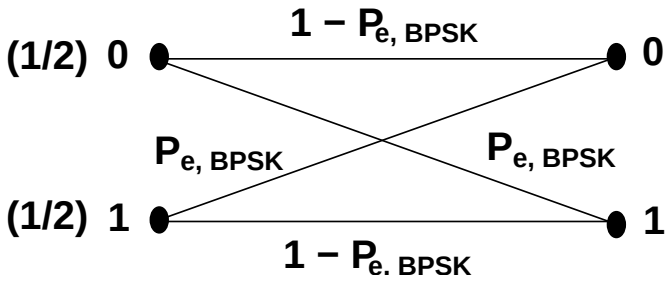


Figure 8: DMC for BPSK signaling.

is defined to be $K' = K(t)$ and the output alphabet size is $J' = J(t)$. This time-variant DMC can be diagrammed as shown in Figure 9.

So far our time-varying system setup is rather general—a sequence of DMCs—and has not been explicitly connected to a model for the channel evolution or to a model for system-parameter selection. First, let's model the channel. Then we will connect the channel to the DMC and finally connect the DMC choice for interval $t+1$ to the DMC and channel estimate for interval t.

Channel Models.

For each time interval indexed by t, we will assume that the channel remains in a particular state. The channel-state random variable is denoted by S and it takes on the A values of $s_1, s_2, \ldots, s_A$. The channel-state random variable for the tth interval is S_t and we would like to characterize the set $\{S_t\}$ of random variables for all t. We assume that the channel cannot make radical changes over reasonable times (that is, T), and that the channel evolves with primary influence coming from its recent history. Thus, we can model the channel evolution using a first-order Markov process (after Gallager [2]),

$$P(S_t|S_{t-1}, S_{t-2}, \ldots, S_{t-K}) \equiv P(S_t|S_{t-1}), \quad \forall K \geq 1 \text{ and } \forall t.$$

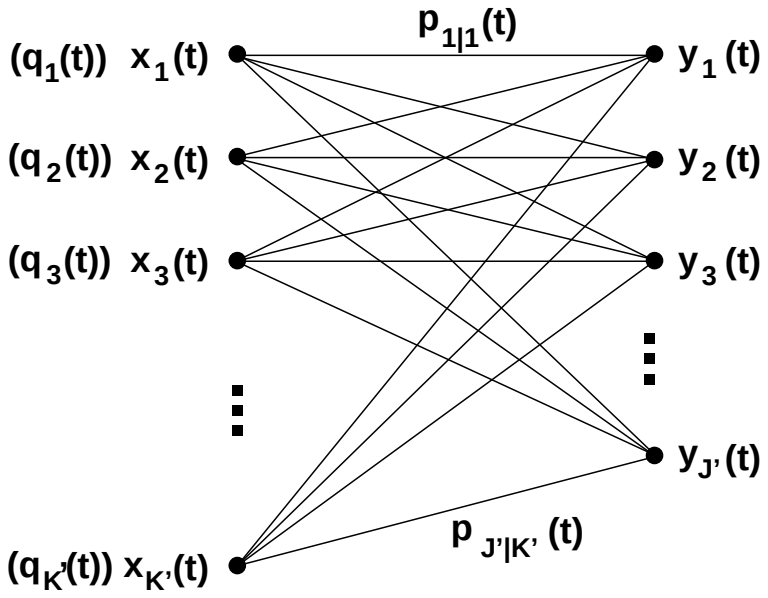


Figure 9: General time-variant DMC for ISP modeling.

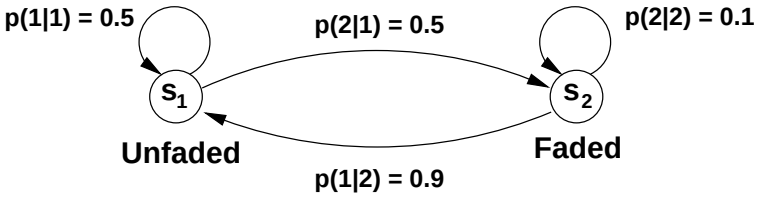
This model does not preclude independent channel states for which

$$P(S_t|S_{t-1}) \equiv P(S_t) \;\; (P(S_t = s_j), j = 1, \ldots, A).$$

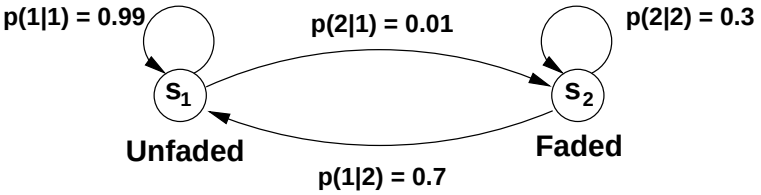
We now present a few simple examples of our channel model.

1. Static AWGN. Let $A = 1$.

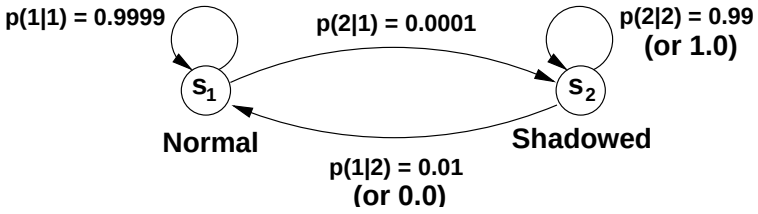
2. Fast-Fading Channels. Let $A = 2$, s_1 denote the unfaded state, and s_2 denote the faded state. A typical state-transition diagram is shown below.



3. Slow-Fading Channels. Let $A = 2$, s_1 denote the unfaded state, and s_2 denote the faded state. A typical state-transition diagram is shown below.



4. Shadowed Channels. Let $A = 2$, s_1 denote the normal state, and s_2 denote the shadowed state. A typical state-transition diagram is shown below.



The channel state affects a fixed DMC only through the transition probabilities. So, each transition probability becomes an explicit function of the channel state rather than the more general function of t,

$$p_{j|k}(t) \longrightarrow p_{j|k}(S_t),$$

which renders the transition probabilities—and hence the DMC—random variables rather than arbitrary functions of time, as shown in Figure 10.

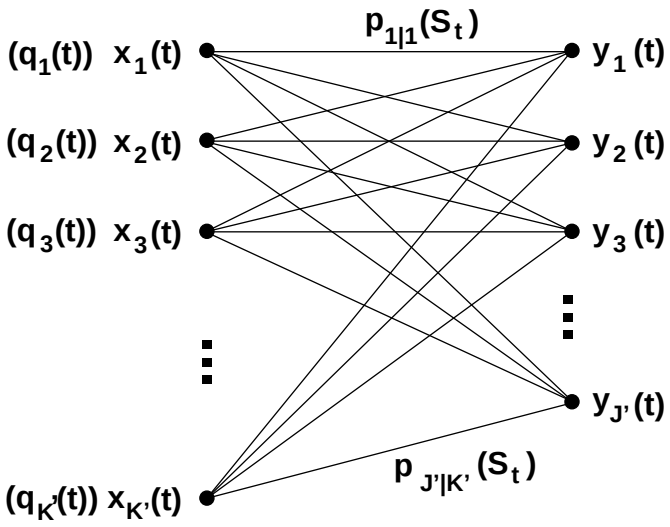


Figure 10: Time-variant channel-dependent DMC for ISP modeling.

DMC Selection based on Channel-State Sensing.

Let us now impose additional structure on the time-variant input and output alphabets and prior probabilities that define a DMC. The central notion is that the system should *sense* the channel state and use that information to select the next parameter set.

Suppose that the system measures the current channel state with some error and arrives at the state estimate random variable Z_t. For each time interval indexed by t, we define the *system* as the input and output alphabets and the prior distribution on the input alphabet. We assume that the destination records all previous channel-state estimates and all previously selected system choices. Let there be B possible systems denoted by $d_1, d_2, \ldots, d_B$ and let the random variable D_t denote the system at interval t. The system D_t is modeled as some function of the current channel-state estimate and all previous system choices,

$$D_{t+1} = f(Z_t, D_t, D_{t-1}, \ldots).$$

Let us further suppose that only the current channel-state estimate and the current system are used to choose the next system,

$$D_{t+1} = f(Z_t, D_t).$$

We can model this probabilistically by another Markov process,

$$P(D_{t+1}|Z_t, D_t, D_{t-1}, \ldots) \equiv P(D_{t+1}|Z_t, D_t).$$

Now, if the destination chooses a system at random without regard to either Z_t or D_t, then

$$P(D_{t+1}|Z_t, D_t) \equiv P(D_{t+1}) \;\; (P(D_{t+1} = d_j), j = 1, \ldots, B).$$

Our DMC diagram now requires three graphs: one for the channel process, one for the system process, and one for the communication process, as shown in Figure 11. To complete our notation, let the channel-state transition probabilities be denoted by $r_{j|a}$,

$$r_{j|a} = P(S_t = s_j|S_{t-1} = s_a),$$

and the system-state transition probabilities be denoted by $v_{j|a,b}$,

$$v_{j|a,b} = P(D_t = d_j|S_{t-1} = s_a, D_{t-1} = d_b).$$

Special Cases of the General ISP Communication-System Model.
Here we point out some special cases of our ISP model having practical or theoretical interest.

1. **Classic DMC on a Static Channel.**
   Here $A = B = 1$, meaning that the communication system is fixed and the channel has a single state. The alphabets, priors, and transition probabilities are time-invariant. This DMC applies to many of the conventional results on communications systems.

2. **Classic DMC on a Time-Variant Channel.**
   Here $A > 1$ and $B = 1$, meaning that the alphabets and priors are fixed, but the channel transition probabilities are time-varying [2] due to the evolving channel state.

3. **Variable DMC on a Static Channel.**
   Here $A = 1$ and $B > 1$, meaning that the channel has a single state and the DMC is time-varying. (This case may have little practical value.)

4. **Uncorrelated DMC and Channel.**
   Here the sequence of DMC systems $\{D_t\}$ is uncorrelated with the sequence of channel-state estimates $\{Z_t\}$. For example, the system does not use measurements to select the DMC, but merely selects it at random. Alternatively, the DMC selection is influenced by time-varying source data-rate constraints.

5. **Unconstrained DMC Selection.**
   Here the system Markov process characterized by $P(D_t|Z_{t-1}, D_{t-1})$ degenerates to $P(D_t|Z_{t-1})$. That is, the choice of the system parameters is not influenced by the current system, only by the current channel-state estimate.

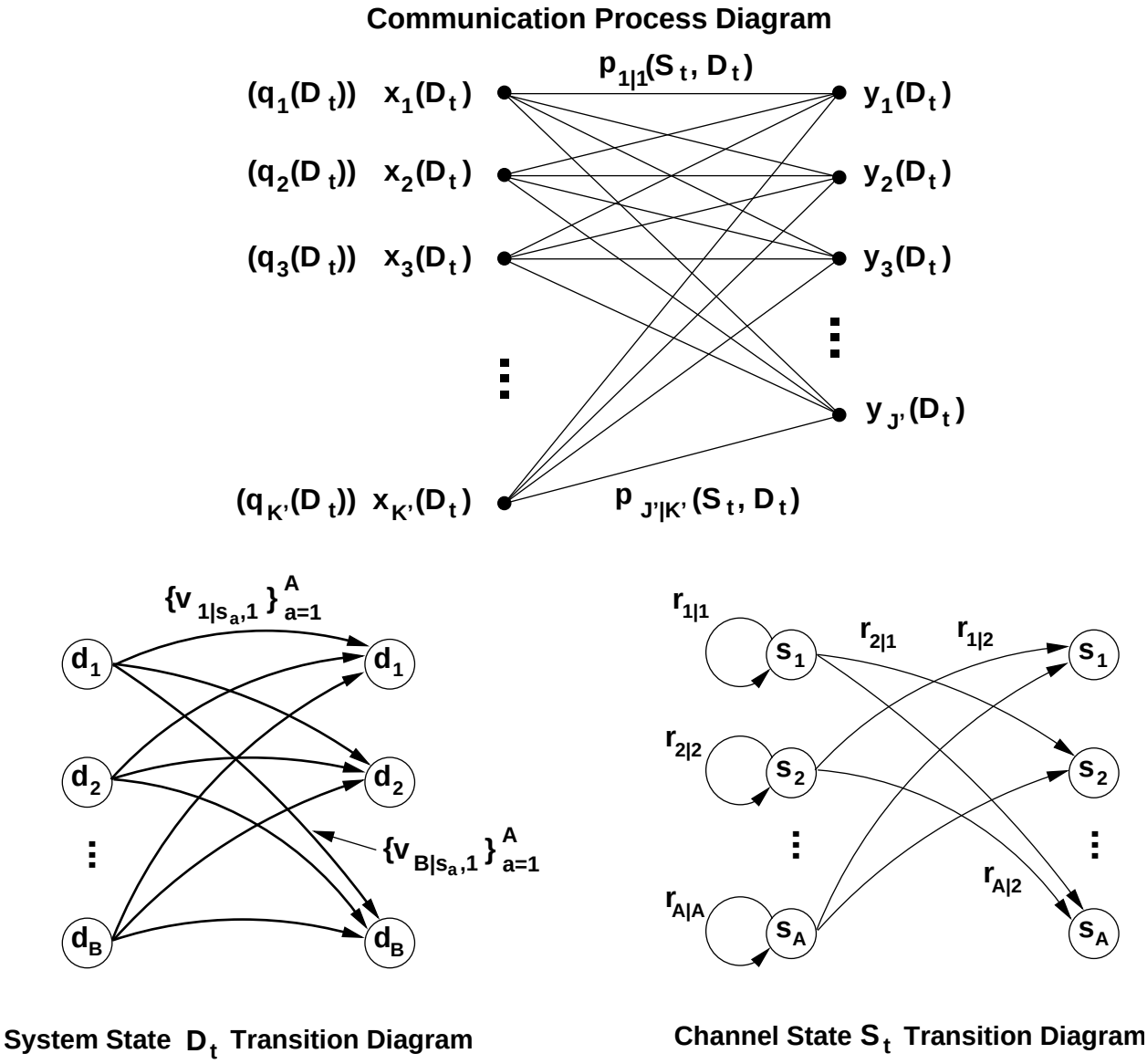


Figure 11: Final diagram for time-variant DMC for ISP modeling.

6. **Constrained DMC Selection.**
   Here the system and channel are first-order Markov processes. The choice of system parameters is explicitly influenced by the previous system and the current channel-state.

Connection to AISPS and CISPS Systems in Section 3.

For the AISPS example in Section 3.1 in which the destination samples performance at each of a set of perturbed positions and selects the best as the new destination position, we have an instance of special case 5 above. The channel-state estimate in this situation is the collection of quality measures indexed by the position vector.

For the random cooperative parameter adjustment CISPS example, we have an instance of special case 4, and for the preset cooperative parameter adjustment example, we have an instance of special case 6. Finally, for the optimal cooperative parameter adjustment example, we have another instance of special case 5.

5 Specifi c Mathematical Problems of Interest

Now that we have developed a general framework for specifying mathematical models for a wide class of ISP-enabled communication systems, we would like to analyze the models to determine their structure and their performance.

The central question before us is: *What is the benefit of an ISP communication system relative to a conventional system?* We desire to pose and solve problems that help answer this question.

Communication systems are judged by five criteria: optimality, error performance, and power, bandwidth, and computational efficiency. For the first of these, the best possible system would transmit at a rate very near the Shannon capacity with excellent error performance. What we would like to do here is find a class of ISP-enabled systems that have a much larger capacity than their traditional (conventional) counterparts, then find engineering solutions that approximate these systems with reasonable power, bandwidth, and computational complexity at a high performance level. On the other hand, large capacity gains may not be found using the present formulation without mandatory order-of-magnitude increases in bandwidth, power, or complexity.

So the first step in our mathematical analysis of the proposed ISP communication system models is finding their capacities [2]–[16] since this will guide us to the situations with highest potential ISP payoff. We include in the problem statements below several baseline problems that also serve as warm-up exercises.

Problem Statements.

1. For an AWGN channel, derive equivalent DMCs for MFSK, MPSK, and MQAM. Include also DSSS BPSK and FH signals if time permits.

2. Given a general time-invariant DMC, derive a formula for capacity. Evaluate the formula for a variety of parameters such as K, J, and the transition probabilities. Make sure results default to known or published results.

3. Given a static system D and a variable channel with $A > 1$ states and a Markov probability structure, derive a formula for capacity. Evaluate numerically and compare to static-channel

baseline. This problem deals with a fixed communication system facing a time-varying channel.

4. Given a static channel S and a variable DMC with $B > 1$ possible system states and a memoryless system state sequence (systems are chosen independent of the channel state estimates with some probability distribution over the B systems), derive capacity and evaluate numerically. This problem deals with a time-varying communication system in the presence of a static channel.

5. For a model employing a variable channel ($A > 1$) and a variable DMC ($B > 1$), assume that the best system is always chosen for the current estimated channel state. Consider two cases: (1) the channel state is estimated perfectly and (2) the channel state is estimated with a known error rate. Derive and evaluate capacity.

6. Given $A > 1$, $B > 1$, and D_t and S_t first-order Markov with each system state d_j connected to at most itself and two additional states, derive and evaluate capacity.

6 Engineering Issues

We expect that if any large capacity or performance gains are predicted by the mathematical analysis of our ISP model, they will require relatively sophisticated implementations. In particular, the ability to radically and quickly alter modulation type will be greatly enhanced by the use of the emerging software-radio and high-speed computational technologies. Similarly, the ability to radically and quickly modify the carrier frequency across a wide band will be enhanced by cutting-edge switching and oscillator technology.

Significant engineering issues are expected to center on how best to design any required periodically transmitted pilot symbols and on the specific elements that enter the modifiable-parameter set P_a.

7 Problem Analysis

In this section we present our analysis results for the problems posed in Section 5.

7.1 Formal Definitions

Definition 1 (Input Alphabet) *Let the input alphabet for a static (time-invariant) discrete memoryless channel be defined by the K numbers $x_1, \ldots, x_K$. For a dynamic (time-varying) DMC, the size of the alphabet is variable. For the ith DMC D_i, the alphabet size is $K(D_i) = K'$.*

Definition 2 (Output Alphabet) *Let the output alphabet of a static DMC be defined by the J numbers $y_1, \ldots, y_J$. For the ith DMC D_i in a sequence of DMCs, the size is a function of D_i: $J' = J(D_i)$.*

Definition 3 (Prior Input Probabilities) *Let the prior probabilities on the input alphabet for DMC D_i be denoted by $q_1(i, D_i), \ldots, q_{K(D_i)}(i, D_i)$. For each i we require*

$$\sum_{k=1}^{K(D_i)} q_k(i, D_i) = 1.$$

For a static DMC, the prior probabilities are denoted simply by q_k, $k = 1, \ldots, K$.

Definition 4 (Transition Probabilities) *The transition probabilities for a static DMC with K input letters and J output letters is defined by*

$$P(\mathsf{y} = y_j | \mathsf{x} = x_k) = P(j|k) = p_{j|k},$$

for $k = 1, \ldots, K$ and $j = 1, \ldots, J$. For the ith DMC D_i in a sequence of DMCs and a dynamic channel with channel state S_i, we have

$$P(\mathsf{y}_i = y_j | \mathsf{x}_i = x_k, S_i, D_i) = P(j|k, S_i, D_i) = p_{j|k}(S_i, D_i).$$

When the channel is static the conditional transition probabilities reduce to $p_{j|k}(D_i)$ and when the DMC is fixed they reduce to $p_{j|k}(S_i)$.

Definition 5 (Dynamic Channel Model) *The dynamic channel is modeled as a first-order Markov process with A states $s_1, \ldots, s_A$. The channel state for the ith time interval is modeled by the random variable S_i. The channel-state transition probabilities obey the relation*

$$\begin{aligned} P(S_{i+1} = s_{j_1} | S_i = s_{j_2}, S_{i-1} = s_{j_3}, \ldots) &= P(S_{i+1} | S_i, S_{i-1}, \ldots) \\ &= P(S_{i+1} | S_i), \end{aligned}$$

for all $i, j_1, j_2, \ldots$.

Definition 6 (System Parameter Set) *A* system *is defined by the selection of an input alphabet, output alphabet, and prior probabilities on the input alphabet. A system is typically denoted by D.*

Definition 7 (Channel Estimator Model) *The channel estimate for time interval i is denoted by Z_i; this is an estimate of S_i and can take on the values $s_1, \ldots, s_A$. The estimator is a function of the current channel state only and is modeled probabilistically as*

$$P(Z_i = s_{j_1} | S_i = s_{j_2}) = P(Z_i | S_i).$$

For perfect channel-state estimation, $P(Z_i = s_{j_1} | S_i = s_{j_2})$ is 1 for $j_1 = j_2$ and is zero otherwise.

Definition 8 (Dynamic System Model) *The system evolves as a function of previous systems and the previous channel-state estimates. The system in the ith time interval is denoted by D_i and can take on one of the B values $d_1, \ldots, d_B$. The system sequence is modeled as a Markov process,*

$$\begin{aligned} P(D_{i+1} = d_l | S_i = s_{j_1}, S_{i-1} = s_{j_2}, \ldots, D_i = d_{k_1}, D_{i-1} = d_{k_2}, \ldots) &= P(D_{i+1} = d_l | D_i = d_{k_1}, \\ &\quad S_i = s_{j_1}) \\ &= P(D_{i+1} | S_i, D_i), \end{aligned}$$

Definition 9 (Channel-Use Sequence) *The random variables* x *and* y *denote channel inputs and outputs. Thus,* x *takes on the values* $x_1, \ldots, x_K$ *and* y *takes on the values* $y_1, \ldots, y_J$*. Consider a sequence of* N *successive channel uses. Let the* N*-vectors* $\boldsymbol{x} = [\mathsf{x}_1, \ldots, \mathsf{x}_N]$ *and* $\boldsymbol{y} = [\mathsf{y}_1, \ldots, \mathsf{y}_N]$ *denote the random inputs and outputs, respectively. The* i*th element of* $\boldsymbol{x}$*,* x_i*, takes one of the values* $x_1, \ldots, x_{K(D_i)}$ *and similarly* y_i *takes one of the values* $y_1, \ldots, y_{J(D_i)}$*. Let* $\boldsymbol{u} = [u_1, \ldots, u_N]$ *and* $\boldsymbol{w} = [w_1, \ldots, w_N]$ *denote valid values for* $\boldsymbol{x}$ *and* $\boldsymbol{y}$*, respectively. The sequence of channel states is* $\boldsymbol{S} = [S_1, \ldots, S_N]$*, where* S_i *can take on the values* $s_1, \ldots, s_A$*. Similarly, the sequence of systems is* $\boldsymbol{D} = [D_1, \ldots, D_N]$ *where* $D_i \in \{d_1, \ldots, d_B\}$*. Let* $\boldsymbol{\sigma} = [\sigma_1, \ldots, \sigma_N]$ *and* $\boldsymbol{\delta} = [\delta_1, \ldots, \delta_N]$ *denote valid values for* $\boldsymbol{S}$ *and* $\boldsymbol{D}$*, respectively.*

Definition 10 *Symmetric Channel A DMC is symmetric if the set of outputs can be partitioned into subsets in such a way that for each subset the matrix of transition probabilities has the property that each row is a permutation of each other row and each column is a permutations of each other column [2].*

7.2 Problem 1: Equivalent DMCs for Various Modulation Types

This aspect of the work was deferred in favor of the other work reported on herein, and due to changes in contract funding and objectives, we were unable to return to it.

7.3 Problem 2: Static DMC and Channel

Theorem 1 (Mutual Information for Static DMC and Static Channel)
The average mutual information for a static system on a static channel is given by

$$I(X;Y) = \sum_{j=1}^{J} \sum_{k=1}^{K} p_{j|k} q_k \log \left(\frac{p_{j|k}}{\sum_{l=1}^{K} q_k p_{j|l}} \right).$$

Theorem 2 (Capacity for a Static DMC and Static Channel)
The capacity of a static system on a symmetric static channel is achieved by using equiprobable input letters, and is given by

$$\begin{aligned} C &= \max_{\{q_k\}} I(X;Y) \\ &= \frac{1}{K} \sum_{j,k} p_{j|k} \log \left(\frac{K p_{j|k}}{\sum_{l=1}^{K} p_{j|l}} \right). \end{aligned}$$

7.4 Problem 3: Static DMC and Dynamic Channel

Theorem 3 (Mutual Information for Static DMC and Dynamic Channel)
The average mutual information for a static system facing a dynamic channel over N *channel uses is given by*

$$I(X^N;Y^N) = \sum_{\boldsymbol{u},\boldsymbol{w}} P(\boldsymbol{y} = \boldsymbol{w} | \boldsymbol{x} = \boldsymbol{u}) P(\boldsymbol{x} = \boldsymbol{u}) \log \left(\frac{P(\boldsymbol{y} = \boldsymbol{w} | \boldsymbol{x} = \boldsymbol{u})}{P(\boldsymbol{y} = \boldsymbol{w})} \right),$$

where

$$P(\boldsymbol{y}=\boldsymbol{w}|\boldsymbol{x}=\boldsymbol{u}) = \sum_{\boldsymbol{\sigma}} P(S_1=\sigma_1)\prod_{j=1}^{N-1} P(S_{j+1}=\sigma_{j+1}|S_j=\sigma_j) \times\left[\prod_{k=1}^{N} P(\mathsf{y}_k=w_k|\mathsf{x}_k=u_k,S_k=\sigma_k)\right],$$

$$P(\boldsymbol{y}=\boldsymbol{w}) = \sum_{\boldsymbol{\sigma}}\left[P(S_1=\sigma_1)\prod_{j=1}^{N-1} P(S_{j+1}=\sigma_{j+1}|S_j=\sigma_j)\right]\left[\prod_{k=1}^{N} P(\mathsf{y}_k=w_k|S_k=\sigma_k)\right],$$

$$P(\mathsf{y}_k=w_k|S_k=\sigma_k) = \sum_{l=1}^{K} P(\mathsf{y}_k=w_k|\mathsf{x}_k=x_l,S_k=\sigma_k)q_l,$$

and

$$P(\boldsymbol{x}=\boldsymbol{u}) = \prod_{j=1}^{N} q_{i_j}, \quad u_j = x_{i_j}.$$

Theorem 4 (Capacity for a Static DMC and Dynamic Channel)
The capacity of a static system on a symmetric dynamic channel is achieved by using equiprobable inputs for each system, and is given by

$$\begin{aligned} C &= \lim_{N\to\infty}\max_{\{q_k\}}\frac{1}{N}I(X^N;Y^N) \\ &= \lim_{N\to\infty}\frac{1}{N}\,I(X^N;Y^N)\big|_{q_l=1/K}. \end{aligned}$$

7.5 Problem 4: Dynamic DMC and Static Channel

Theorem 5 (Mutual Information for Dynamic DMC and Static Channel)
The average mutual information for a dynamic system facing a static channel over N channel uses is given by

$$I(X^N;Y^N) = \sum_{\boldsymbol{u},\boldsymbol{w}} P(\boldsymbol{y}=\boldsymbol{w}|\boldsymbol{x}=\boldsymbol{u})P(\boldsymbol{x}=\boldsymbol{u})\log\left(\frac{P(\boldsymbol{y}=\boldsymbol{w}|\boldsymbol{x}=\boldsymbol{u})}{P(\boldsymbol{y}=\boldsymbol{w})}\right),$$

where

$$P(\boldsymbol{y}=\boldsymbol{w}|\boldsymbol{x}=\boldsymbol{u}) = \sum_{\boldsymbol{\delta}} P(D_1=\delta_1)\prod_{j=1}^{N-1} P(D_{j+1}=\delta_{j+1}|D_j=\delta_j) \times\left[\prod_{k=1}^{N} P(\mathsf{y}_k=w_k|\mathsf{x}_k=u_k,D_k=\delta_k)\right],$$

$$P(\boldsymbol{y}=\boldsymbol{w}) = \sum_{\boldsymbol{\delta}} P(D_1=\delta_1)\prod_{j=1}^{N-1} P(D_{j+1}=\delta_{j+1}|D_j=\delta_j) \times \left(\prod_{i=1}^{N}\sum_{l=1}^{K(\delta_i)} q_l(i,\delta_i)P(\mathsf{y}_i=w_i|\mathsf{x}_i=x_l,D_i=\delta_i)\right),$$

and

$$P(\boldsymbol{x}=\boldsymbol{u}) = \sum_{\boldsymbol{\delta}} P(\boldsymbol{D}=\boldsymbol{\delta})\prod_{j=1}^{N} q_l(j,\delta_j), \quad u_j = x_l.$$

Theorem 6 (Mutual Information for Random DMC and Static Channel)
The average mutual information for a randomly selected DMC on a static channel over N channel uses is given by

$$I(X^N;Y^N) = \sum_{\boldsymbol{u},\boldsymbol{w}} P(\boldsymbol{y}=\boldsymbol{w}|\boldsymbol{x}=\boldsymbol{u})P(\boldsymbol{x}=\boldsymbol{u})\log\left(\frac{P(\boldsymbol{y}=\boldsymbol{w}|\boldsymbol{x}=\boldsymbol{u})}{P(\boldsymbol{y}=\boldsymbol{w})}\right),$$

where

$$P(\boldsymbol{y}=\boldsymbol{w}|\boldsymbol{x}=\boldsymbol{u}) = \sum_{\boldsymbol{\delta}}\prod_{j=1}^{N} P(D_j=\delta_j)\left[\prod_{k=1}^{N} P(\mathsf{y}_k=w_k|\mathsf{x}_k=u_k,D_k=\delta_k)\right],$$

$$P(\boldsymbol{y}=\boldsymbol{w}) = \sum_{\boldsymbol{\delta}}\left(\prod_{j=1}^{N} P(D_j=\delta_j)\right)\left(\prod_{i=1}^{N}\sum_{l=1}^{K(\delta_i)} q_l(i,\delta_i)P(\mathsf{y}_i=w_i|\mathsf{x}_i=x_l,D_i=\delta_i)\right),$$

and

$$P(\boldsymbol{x}=\boldsymbol{u}) = \sum_{\boldsymbol{\delta}}\left(\prod_{k=1}^{N} P(D_k=\delta_k)\right)\left(\prod_{j=1}^{N} q_l(j,\delta_j)\right), \quad u_j = x_l.$$

Proposition 1 (Capacity for a Dynamic DMC and Static Channel)
The capacity of a dynamic system on a symmetric static channel is achieved by using equiprobable inputs for each system, and is given by

$$\begin{aligned} C &= \lim_{N\to\infty}\max_{\{q_k\}}\frac{1}{N}I(X^N;Y^N) \\ &= \lim_{N\to\infty}\frac{1}{N}\,I(X^N;Y^N)\big|_{q_l(j,\delta_j)=1/K(\delta_j)}. \end{aligned}$$

7.6 Problems 5 and 6: Dynamic DMC and Dynamic Channel

Theorem 7 (Mutual Information for Dynamic DMC and Dynamic Channel)
The average mutual information for a dynamic system with channel-state estimation facing a dynamic channel over N channel uses is given by

$$I(X^N;Y^N) = \sum_{\boldsymbol{u},\boldsymbol{w}} P(\boldsymbol{y}=\boldsymbol{w}|\boldsymbol{x}=\boldsymbol{u})P(\boldsymbol{x}=\boldsymbol{u})\log\left(\frac{P(\boldsymbol{y}=\boldsymbol{w}|\boldsymbol{x}=\boldsymbol{u})}{P(\boldsymbol{y}=\boldsymbol{w})}\right),$$

where

$$P(\boldsymbol{y}=\boldsymbol{w}|\boldsymbol{x}=\boldsymbol{u})=\sum_{\boldsymbol{\delta},\boldsymbol{\sigma}}P(\boldsymbol{y}=\boldsymbol{w}|\boldsymbol{x}=\boldsymbol{u},\boldsymbol{D}=\boldsymbol{\delta},\boldsymbol{S}=\boldsymbol{\sigma})P(\boldsymbol{D}=\boldsymbol{\delta}|\boldsymbol{S}=\boldsymbol{\sigma})P(\boldsymbol{S}=\boldsymbol{\sigma}),$$

$$P(\boldsymbol{y}=\boldsymbol{w}|\boldsymbol{x}=\boldsymbol{u},\boldsymbol{D}=\boldsymbol{\delta},\boldsymbol{S}=\boldsymbol{\sigma})=\prod_{j=1}^{N}P(\mathsf{y}_j=w_j|\mathsf{x}_j=u_j,D_j=\delta_j,S_j=\sigma_j),$$

$$P(\boldsymbol{D}=\boldsymbol{\delta}|\boldsymbol{S}=\boldsymbol{\sigma})=P(D_1=\delta_1)\prod_{j=1}^{N-1}\sum_{l\in\mathcal{A}}P(D_{j+1}=\delta_{j+1}|D_j=\delta_j,Z_j=s_l)P(Z_j=s_l|S_j=\sigma_j),$$

$$P(\boldsymbol{S}=\boldsymbol{\sigma})=P(S_1=\sigma_1)\prod_{k=1}^{N-1}P(S_{k+1}=\sigma_{k+1}|S_k=\sigma_k),$$

$$P(\boldsymbol{y}=\boldsymbol{w})=\sum_{\boldsymbol{\delta},\boldsymbol{\sigma}}P(\boldsymbol{D}=\boldsymbol{\delta}|\boldsymbol{S}=\boldsymbol{\sigma})P(\boldsymbol{S}=\boldsymbol{\sigma})\left(\prod_{i=1}^{N}\sum_{l=1}^{K(\delta_i)}q_l(i,\delta_i)P(\mathsf{y}_i|x_l,\delta_i,\sigma_i)\right),$$

$$P(\boldsymbol{x}=\boldsymbol{u})=\sum_{\boldsymbol{\sigma},\boldsymbol{\delta}}P(\boldsymbol{D}=\boldsymbol{\delta}|\boldsymbol{S}=\boldsymbol{\sigma})P(\boldsymbol{S}=\boldsymbol{\sigma})\prod_{j=1}^{N}q_l(j,\delta_j),\quad u_j=x_l,$$

and $\mathcal{A}=\{1,\ldots,A\}$.

Remark 1 *Whenever the system is dynamic, there are several possible capacity definitions. Each definition involves a different maximization of average mutual information. In the most general case, we maximize over the entire collection of priors* $\{q_l(i,D_i)\}$*, a set of size* $N\sum_{k=1}^{B}K(d_k)$*. In a less general case, the priors for all instances of the system* δ_i *are equal and we maximize over the* B *sets of priors, which is a smaller set of size* $\sum_{k=1}^{B}K(d_k)$*.*

Proposition 2 (Capacity for a Dynamic DMC and Dynamic Channel)
The capacity of a dynamic system on a symmetric dynamic channel is achieved by using equiprobable inputs for each system, and is given by

$$\begin{aligned}C &= \lim_{N\to\infty}\max_{\{q_k\}}\frac{1}{N}I(X^N;Y^N)\\ &= \lim_{N\to\infty}\frac{1}{N}\,I(X^N;Y^N)\big|_{q_l(j,\delta_j)=1/K(\delta_j)}.\end{aligned}$$

8 Discussion and Numerical Examples

The formulas for the most general ISP communication link—a dynamic system and dynamic channel—have been coded in MATLAB. Since all the simpler cases are special cases of this formula, all capacity formulas provided in the preceding theorem statements can be evaluated. For even modest problems involving two systems ($B=2$) and two channel states ($A=2$), the capacity formulas are costly to evaluate even for small values of N such as 4 or 5. Nevertheless, we

present here a few examples to illustrate the potential advantages of the ISP approach over static approaches.

Recall that the basic idea behind the present effort is to find simple ISP systems that have relatively large capacity for difficult channels. To this end we will be interested in both the absolute capacity of the ISP (dynamic) link and the ratio of capacities between the dynamic and static links. Before we present initial results in this vein, we first present a set of results aimed at providing verification of the formulas and software.

8.1 Example 1: Verification of Formulas and Software

In this first set of examples, we provide evidence that the obtained formulas and their software implementation provide correct results. Therefore, we focus on static systems and channels, for which capacity results are either known or are obvious.

We are interested in two distinct kinds of transition probability functions. The first, called *flat*, corresponds to the case in which $p_{j|k} = P_e$ for $j \neq k$ and $p_{k|k} = 1 - (K-1)P_e$. That is, all errors have equal probability P_e. The second, called *exponential*, assigns progressively smaller probabilities of error to errors involving symbols with increasing distance. That is, $p_{j|k} = P_e^{F(j,k,K)}$, where

$$F(j,k,K) = \min\{|j-k| \mod K,\ ||j-k| - K| \mod K\}.$$

for $j \neq k$, and

$$p_{k|k} = 1 - \sum_{j \neq k} p_{k|j}.$$

Notice that the flat and exponential error models are identical for binary DMCs.

The first result corresponds to a binary DMC facing a static channel. The software is used to compute the capacity of the link as a function of the cross-over (transition) error probability P_e (see Figure 8). The resulting capacity is plotted in Figure 12, which can be compared to results in many communication and information-theory textbooks. The capacity has a maximum of one bit per channel use, which is intuitively obvious, and a minimum of zero bits when the cross-over probability reaches 0.5.

The second result corresponds to various Kary DMCs facing static flat and exponential channels. The capacities are plotted in Figure 13. For the flat channel, notice that the capacities for each K are equal to $\log_2(K)$ bits per channel use when the channel is perfect ($P_e = 0$). Then each capacity reaches zero when $P_e = 1/\log_2(K)$, indicating that all probabilities in the transition diagram are equal which is, again, intuitively pleasing. For the exponential channel, the capacities decrease more slowly with increasing P_e, as can be expected.

8.2 Example 2: Binary Systems Facing a Two-State Channel

In this second example, we consider a dynamic system consisting of two binary DMCs and a dynamic two-state channel. The example is broken into two parts. The first part deals with an interferer that bounces back and forth between the two systems' bands, and the second part deals with a random, long, deep fade (or shadow).

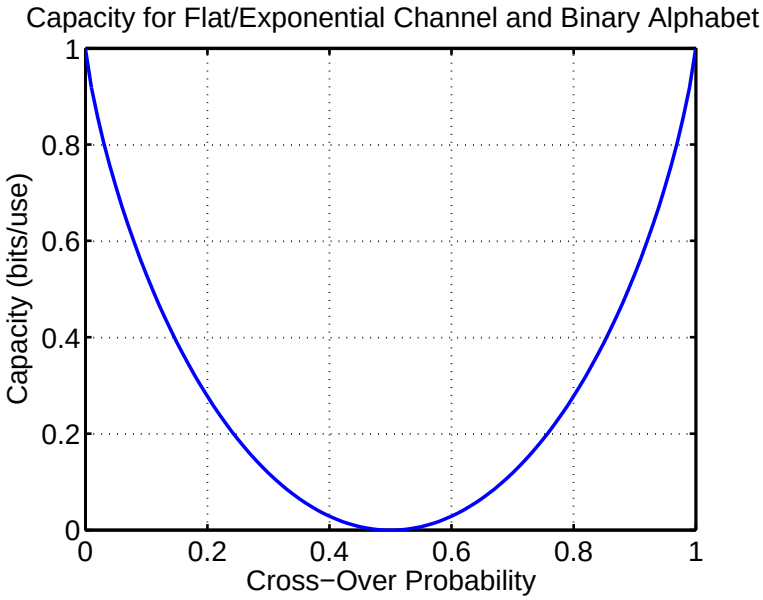


Figure 12: Computed capacity for a binary DMC facing a static channel.

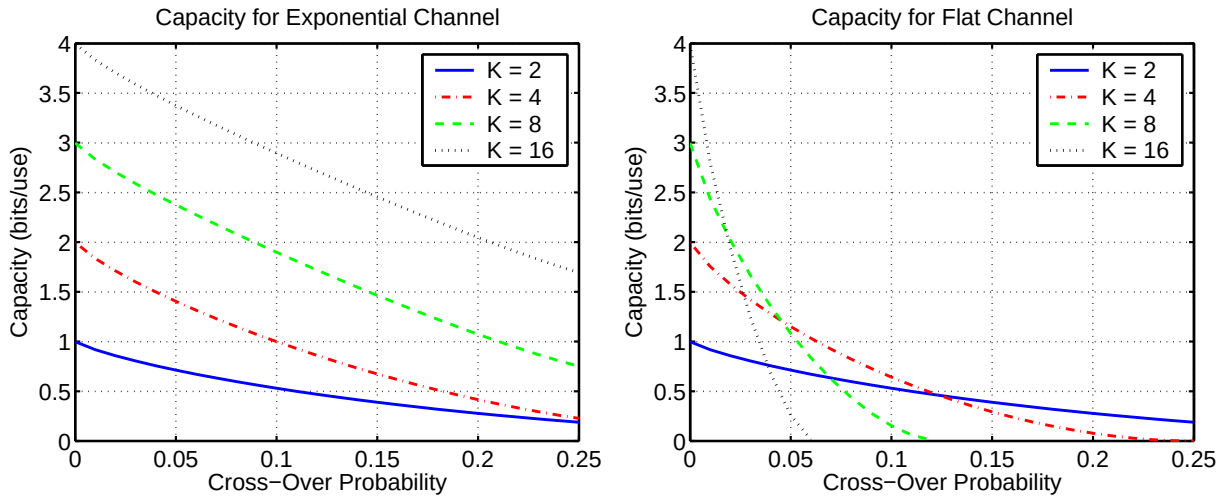


Figure 13: Computed capacities for DMCs with Kary alphabets in flat and exponential noise channels.

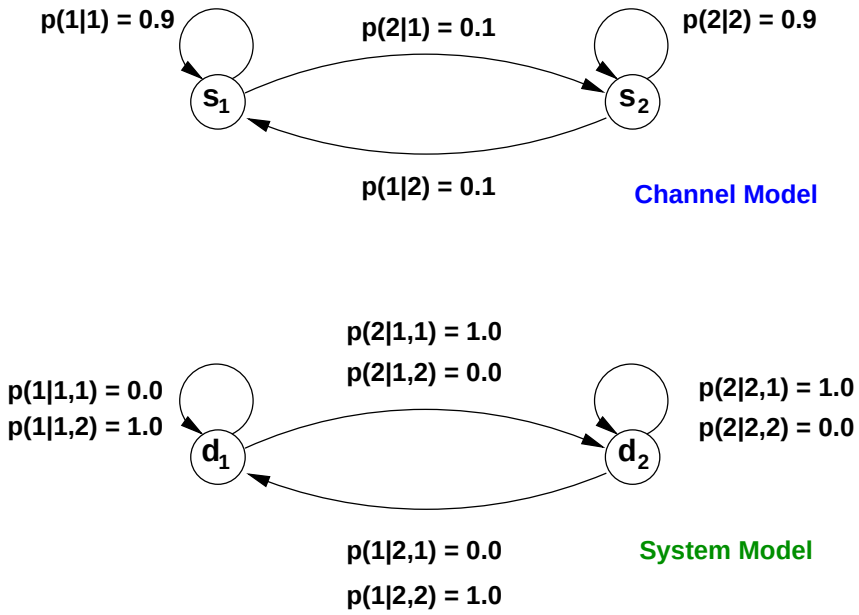


Figure 14: Channel and system state diagrams for the first case in Example Two.

Random Interferer or Jammer.

The basic notion here is that the two DMCs correspond to binary links operating in disjoint frequency bands. During any one time interval, the interferer is in only one of the bands. This notion leads to the channel and system evolution models diagrammed in Figure 14. The nominal transition probabilities for each system-channel combinations are as follows

System	Channel	P_e
1	1	0.5
1	2	10^{-6}
2	1	10^{-6}
2	2	0.5

That is, when the channel state is equal to the system index, the interferer is in that system's band and communication is not possible. Thus, the system evolution model in Figure 14 indicates that when the channel-state estimate is 1 and the current system index is 1, always switch to system 2. Similarly, switch from system 2 to system 1 when the channel-state estimate is 2. Finally, perfect (error-free) channel-state estimation is assumed in this example.

We first compute the capacity for system 1 facing the two-state channel. Computational costs limit the number of channel uses in the calculations to eight. The capacity is computed as a function of the transition (cross-over) probabilities for the two system-channel combinations: $p_1 = P_e$ for channel state 1 and $p_2 = P_e$ for channel state 2. The capacity is shown on the left in Figure 15. Clearly, the capacity approaches zero when both p_1 and p_2 approach 0.5. The computed capacity of the two-system link is shown on the right in Figure 15. Here the transition probabilities for system two are p_2 for channel state 1 and p_1 for channel state 2. A more revealing look at the results is the ratio of the dynamic capacity to the static capacity, shown in Figure 16. Here it is seen that the dynamic capacity can exceed twice that of the static capacity when p_1 is large and p_2 is relatively

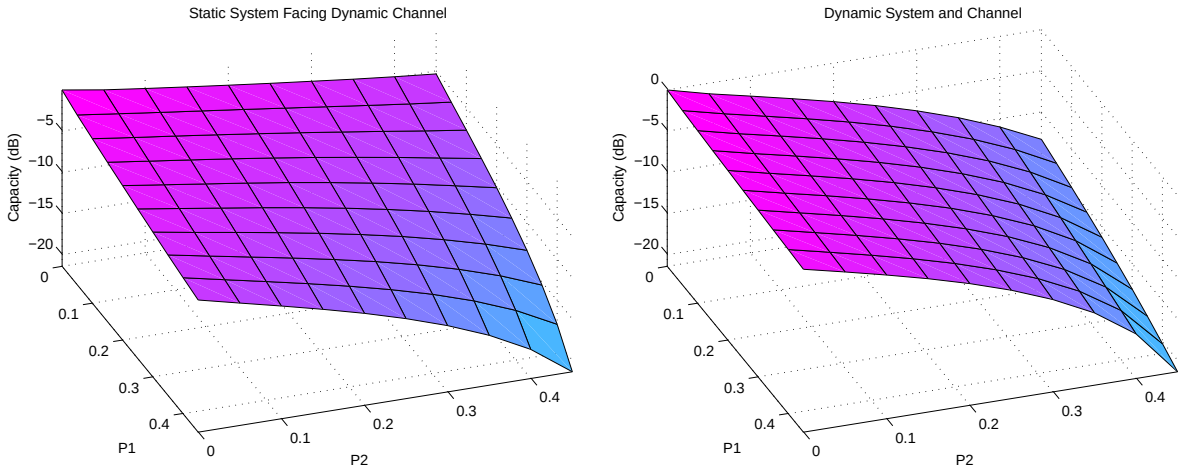


Figure 15: Static and dynamic system capacities for the first case in Example Two.

small. For example, for $p_1 = 0.5$ and $p_2 = 10^{-6}$, we have a static capacity of 0.25 bits per channel use and a dynamic capacity of 0.63.

Random Deep Fade

In the second part of this example, we again consider binary DMCs and a two-state channel, but in this case the channel alternates between a good channel and a deep-fade channel. When the channel is faded, each of the two binary DMCs experience the fade, but system two is much more tolerant to fades that system 1. The basic idea is that the two system operate in overlapping frequency bands, but use different modulation types (say coherent BPSK and incoherent BFSK), which represent widely different combinations of power and bandwidth efficiencies.

The channel and system evolution state diagrams are provided in Figure 17, and the nominal transition probabilities for the four channel-system combinations are as follows

System	Channel	P_e
1	1	10^{-6}
1	2	0.5
2	1	10^{-8}
2	2	0.005

Thus, system 2 is generally superior to system 1 in both channel states, but has some other undesirable qualities such as bandwidth inefficiency. When system 1 encounters a fade, it switches to system 2, which operates until the fade ends.

The capacity for static system 1 and the two-state dynamic channel is shown on the left in Figure 18 as a function of $p_1 = P_e$ for channel state 1 and $p_2 = P_e$ for channel state 2 (for a sequence of eight channel uses). On the other hand, the computed capacity for the dynamic link is shown on the right in Figure 18. Here, the transition probabilities for system 2 are always a factor of 100 smaller than those for system 1. The ratio of dynamic system capacity to static is shown in Figure 19. Note that the ratio is never less than unity and that it gets very large when the transition probabilities for system one both approach 0.5. For the nominal probabilities above (10^{-6} and 0.5), we have the static capacity for system 1 of 0.19 and the dynamic capacity of 0.48. When the fade

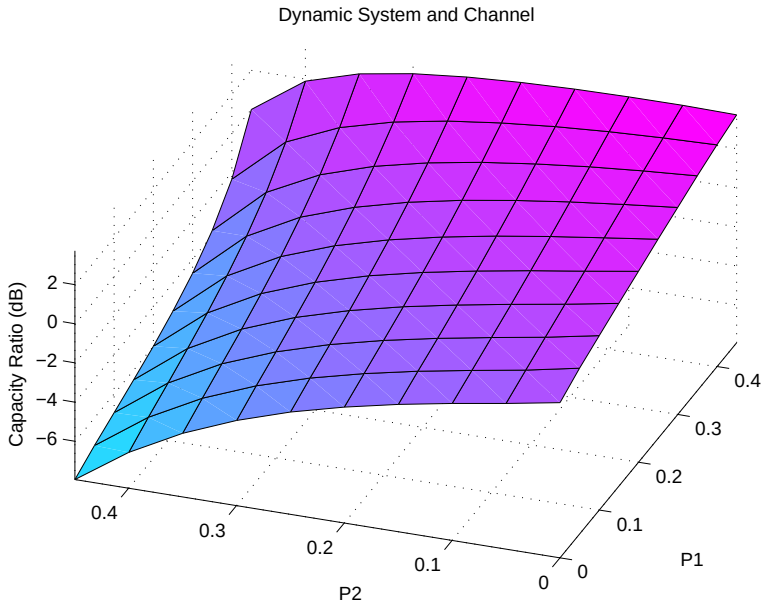


Figure 16: The static-dynamic capacity ratio for the first case in Example Two.

is such that system 1 simply does not work, system 2 still does, and the capacity ratio is very large because the static-system capacity is close to zero.

8.3 Example 3: Mixed-Rate Systems Facing a Multi-State Channel

In this final example, we study the challenging example involving a rate-adaptive system facing a dynamic channel. The intent of the example is to compare a typical rate-adaptive scheme with an ISP-enabled adaptive scheme.

We consider three possible bit rates corresponding to binary, 4-ary, and 8-ary modulation types. For the typical rate-adaptive scheme, signalling is performed in a single fixed frequency band and there are four possible channel states, roughly corresponding to the SNR condition seen by the current demodulator: High, Moderate, Low, and Blocked. For the ISP system, we simply perform the rate-adaptation in one of two possible frequency bands. When one band becomes blocked, the system moves to the other band. We assume that the two bands are never simultaneously Blocked.

Channel-State Transitions.

Since there are two frequency bands and four possible states for each, there are sixteen distinct channel states. We will explicitly rule out the state in which both bands are Blocked, so that the total number of possible states is fifteen, as defined in Table 2.

To define the channel-state transition probabilities, we first assume that the probability of staying in the current state is high. Also, transitions are allowed only between adjacent states. For example, the Band-1 state can transition between High and Moderate, but not between High and Low or High and Blocked. This assumption effectively imposes a slow-variation constraint on the physical channel. Finally, once a band is Blocked, it stays Blocked for some time.

We assign the same-state probability $p(i|i) = 0.91$. All other allowed transitions are equiprobable for each i. For example, for $i = 5$ we have $p(1|5) = p(6|5) = p(9|5) = 0.03$.

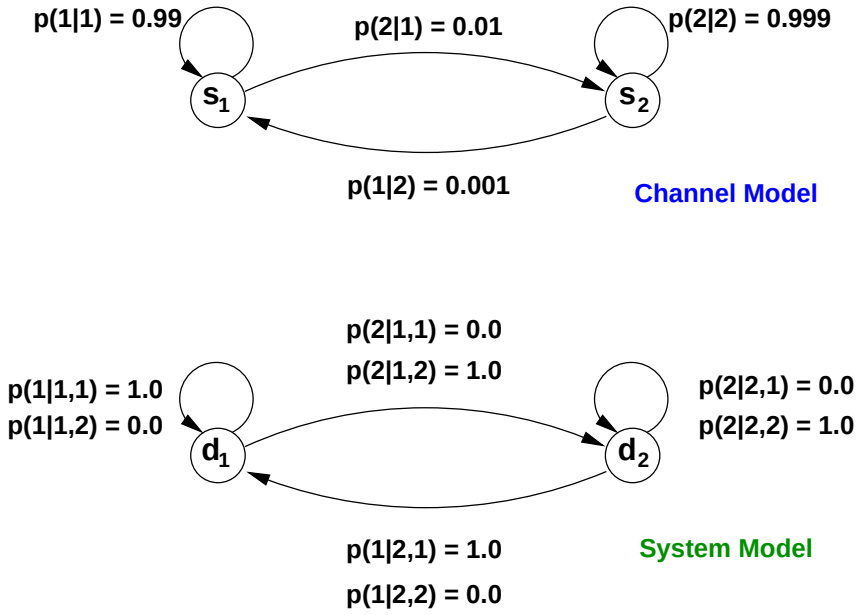


Figure 17: Channel and system state diagrams for the second case in Example Two.

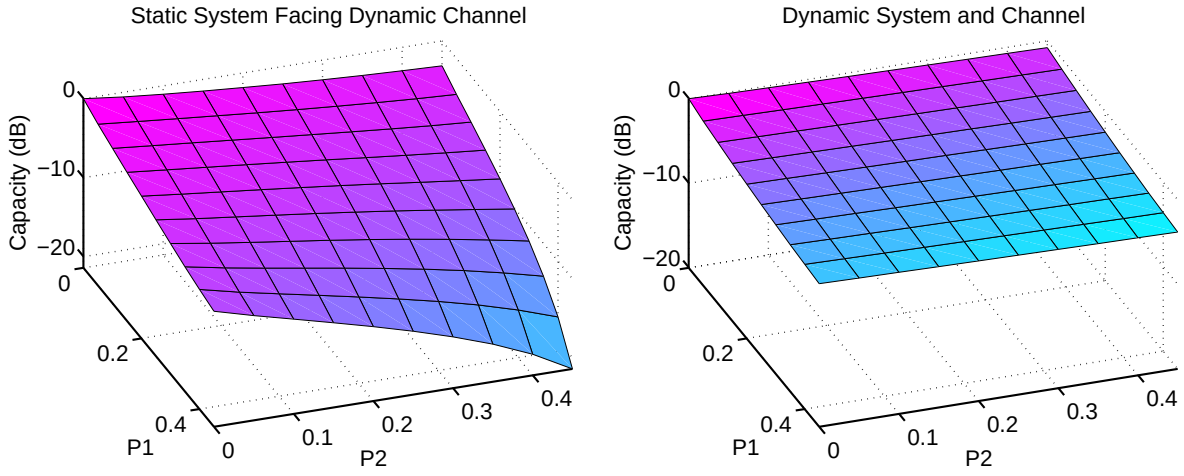


Figure 18: Computed capacities for the links in the second part of Example Two.

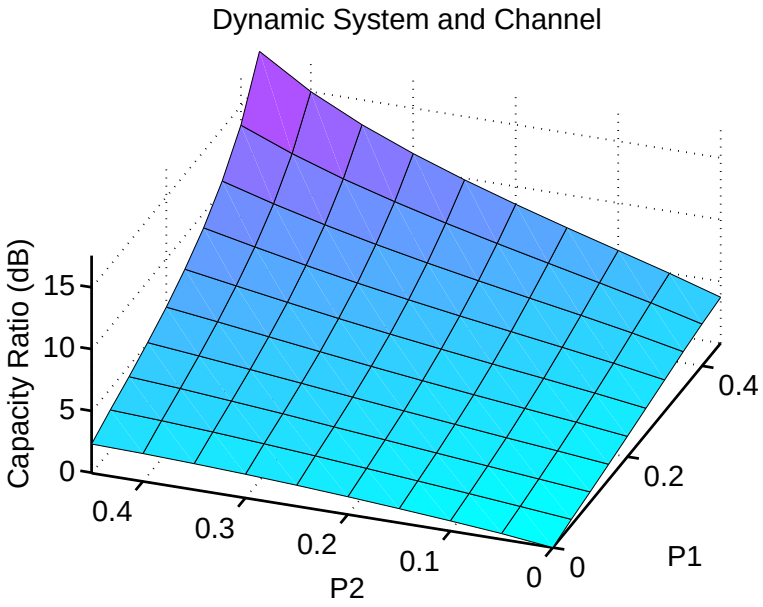


Figure 19: Computed capacity ratio for the second part of Example Two.

State Label	Band 1	Band 2
1	High	High
2	High	Moderate
3	High	Low
4	High	Blocked
5	Moderate	High
6	Moderate	Moderate
7	Moderate	Low
8	Moderate	Blocked
9	Low	High
10	Low	Moderate
11	Low	Low
12	Low	Blocked
13	Blocked	High
14	Blocked	Moderate
15	Blocked	Low

Table 2: Channel-state labels for Example Three.

System Label	Band Label	Modulation M
1	1	2
2	1	4
3	1	8
4	2	2
5	2	4
6	2	8

Table 3: System labels for Example Three.

System Evolution.
To define the system-evolution probabilities, first note that there are six possible systems as shown in Table 3. We need to specify the conditional probabilities

$$P(d_{i+1} = \Delta | d_i = \gamma, s_i = \sigma),$$

where d_{i+1} and d_i denote the next and current systems, respectively, and s_i denotes the current channel state. Our system evolution is governed by the following guidelines. Whenever the current state is High, increase M. Whenever it is Low, decrease M. When it is Moderate, decrease M if M is maximum, increase if M is minimum. If the current channel-state is Blocked, move to the other frequency band and maintain M. This results in a set of system evolution probabilities that are either unity or zero.

Channel-State Estimator.
For this experiment, we assume that the channel state is perfectly estimated. An example estimator involves measuring the tightness of the clusters of received points in the constellation diagram. For very tight clusters, the channel state is estimated as High, for somewhat loose clusters, it is estimated at Moderate, etc.

DMC Transition Probabilities.
The transition probabilities depend on both the current system and the current channel state. Let us assume that the $M = 2, 4,$ and 8 modulation types are well-matched to the Low, Moderate, and High SNR levels. Mismatches between the value of M and the channel state result in penalties of 100.0 and rewards of 0.01, depending on the orientation of the mismatch. The nominal BER for matched situations is 1.0e-4. Thus, when the channel state is Moderate and $M = 4$, the BER is assumed to be 1.0e-4. On the other hand, when the channel state is Moderate and $M = 2$, the BER is 1.0e-6.

Non-ISP Rate-Adaptive System.
For the Non-ISP rate-adaptive system, only one frequency band is available. Thus, there are only four channel states and three systems. If the band becomes Blocked, communication takes place at a BER of 0.5 for all values of M.

Results for Example Three.
For this example, the computational burden is quite high since the dimension of the largest system

	Channel Sequence Length N			
	2	3	4	5
Non-ISP	0.94	1.07	1.11	1.12
ISP	1.63	1.87	1.97	2.00

Table 4: Capacity results for Example Three.

is eight and the channel has many states. Therefore, we are able to compute the exact capacity only for a single set of parameters (as described) and for at most a sequence of five channel uses. In this case, the computed capacities are as shown in Table 4. Many variations are of interest here, but this single case reveals that even for mild channel-state evolutions, the benefits of ISP-enabled modulation adaptation are quite large. We hope to expand upon this example in future work.

9 Conclusions

The general problem of integrating environment sensing with communication-system processing is addressed in this report. Traditional communication-system design focuses on disjoint optimization of the canonical processing blocks in a communication system. The system is deployed and has little ability to adapt to unforeseen conditions. In the present work, the overarching concept is of a communication system that can sense and assess its environment (channel) and use this information to make appropriate changes to one or more system parameters. Such a system can tolerate degraded conditions and take advantage of improved conditions. Our first goal in developing this system notion is the establishment of an abstract system model. The second goal is computation of channel capacities for the model. Progress toward these two fundamental goals is documented herein. Further work will be aimed at development and evaluation of engineering solutions that take advantage of the increased capacity of the new system model.

References

[1] "A Mathematical Methodology for Managing and Integrating Sensors and Processors in Distributed Systems for Radar and Communications," Mission Research Corporation Proposal for the DARPA ISP Program, October 2001.

[2] R. G. Gallager, *Information Theory and Reliable Communication*, Wiley & Sons, New York, 1968.

[3] A. J. Goldsmith and P. P. Varaiya, "Capacity, Mutual Information, and Coding for Finite-State Markov Channels," *IEEE Trans. IT,* Vol. 42, No. 3, pp. 868–886, May 1996.

[4] A. J. Goldsmith and P. P. Varaiya, "Capacity of Fading Channels with Channel Side Information," *IEEE Trans. IT,* Vol. 43, No. 6, pp. 1986–1992, Nov. 1997.

[5] A. Das and P. Narayan, "Capacities of Time-Varying Multiple-Access Channels with Side Information," *IEEE Trans. IT,* Vol. 48, No. 1, pp. 4–25, Jan. 2002.

[6] A. Lapidoth and P. Narayan, "Reliable Communication Under Channel Uncertainty," *IEEE Trans. IT,* Vol. 44, No. 6, pp. 2148–2177, Oct. 1998.

[7] I. Csiszar and P. Narayan, "Capacity and Decoding Rules for Classes of Arbitrarily Varying Channels," *IEEE Trans. IT,* Vol. 35, No. 4, pp. 752–769, July 1989.

[8] G. Caire and S. Shamai, "On the Capacity of Some Channels with Channel State Information," *IEEE Trans. IT,* Vol. 45, No. 6, pp. 2007–2019, Sept. 1999.

[9] C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Tech. Journal,* Vol. 27, pp. 379–423, 623–656, July, October 1948.

[10] I. C. Abou-Faycal, M. D. Trott, and S. Shamai, "The Capacity of Discrete-Time Memoryless Rayleigh-Fading Channels," *IEEE Trans. IT,* Vol. 47, No. 4, pp. 1290–1301, May 2001.

[11] S. Verdu, "Fifty Years of Shannon Theory," *IEEE Trans. IT,* Vol. 44, No. 6, pp. 2057–2078, Oct. 1998.

[12] A. R. Calderbank, "The Art of Signaling: Fifty Years of Coding Theory," *IEEE Trans. IT,* Vol. 44, No. 6, pp. 2561–2595, Oct. 1998.

[13] K. Yanagi, "An Upper Bound to the Capacity of Discrete Time Gaussian Channel with Feedback–Part II," *IEEE Trans. IT,* Vol. 40, No. 2, pp. 588–593, Mar. 1994.

[14] K. Yanagi, "On the Capacity of the Discrete-Time Gaussian Channel with Delayed Feedback," *IEEE Trans. IT,* Vol. 41, No. 4, pp. 1051–1059, July 1995.

[15] H. W. Chen and K. Yanagi, "Upper Bounds on the Capacity of Discrete-Time Blockwise White Gaussian Channels with Feedback," *IEEE Trans. IT,* Vol. 46, No. 3, pp. 1125–1131, May 2000.

[16] F. Alajaji and N. Whalen, "The Capacity-Cost Function of Discrete Additive Noise Channels with and without Feedback," *IEEE Trans. IT,* Vol. 46, No. 3, pp. 1131–1140, May 2000.

[17] W. T. Webb and R. Steele, "Variable Rate QAM for Mobile Radio," *IEEE Trans. Comm.,* Vol. 43, No. 7, pp. 2223–2230, July 1995.

[18] D. L. Goeckel, "Adaptive Coding for Time-Varying Channels Using Outdated Fading Estimates," *IEEE Trans. Comm.,* Vol. 47, No. 6, pp. 844–855, June 1999.

[19] C. Kose and D. L. Goeckel, "On Power Adaptation in Adaptive Signaling Systems," *IEEE Trans. Comm.,* Vol. 48, No. 11, pp. 1769–1773, May 1998.

[20] A. J. Goldsmith and S-G Chua, "Adaptive Coded Modulation for Fading Channels," *IEEE Trans. Comm.,* Vol. 46, No. 5, pp. 595–602, May 1998.

[21] S. T. Chung and A. J. Goldsmith, "Degrees of Freedom in Adaptive Modulation: A Unified View," *IEEE Trans. Comm.,* Vol. 49, No. 9, pp. 1561–1571, Sept. 2001.

[22] C. H. Wong and L. Hanzo, "Upper-Bound Performance of a Wide-Band Adaptive Modem," *IEEE Trans. Comm.*, Vol. 48, No. 3, pp. 367–369, Mar. 2000.

[23] A. J. Goldsmith and S-G Chua, "Variable-Rate Variable-Power MQAM for Fading Channels," *IEEE Trans. Comm.*, Vol. 45, No. 10, pp. 1218–1230, Oct. 1997.

[24] S. Nanda, K. Balachandran, and S. Kumar, "Adaptation Techniques in Wireless Packet Data Services," *IEEE Comm. Mag.*, pp. 54–64, Jan. 2000.

[25] X. Qiu and K. Chawla, "On the Performance of Adaptive Modulation in Cellular Systems," *IEEE Trans. Comm.*, Vol. 47, No. 6, pp. 884–895, June 1999.

[26] E-L Kuan and L. Hanzo, "Burst-by-Burst Adaptive Multiuser Detection CDMA: A Framework for Existing and Future Wireless Standards," *Proc. IEEE*, Vol. 91, No. 2, pp. 278–302, Feb. 2003.

Appendices

A Definitions of Capacity

In this appendix we provide a brief overview of capacity definitions. There is no single universally applicably definition, although the basic idea in all cases is the maximization of average mutual information. We review capacity for discrete memoryless channels (DMCs), discrete finite-state channels (FSCs), discrete-time memoryless channels, and waveform channels. The material in this appendix is based on Gallager [2].

A.1 Discrete Memoryless Channels

The discrete memoryless channel (DMC) is defined by a finite input alphabet, a finite output alphabet, the prior probabilities on the input alphabet, and the transition probabilities, as shown in Figure 20. The transition probability $p_{j|k}$ is the probability of receiving y_j given that x_k is transmitted.

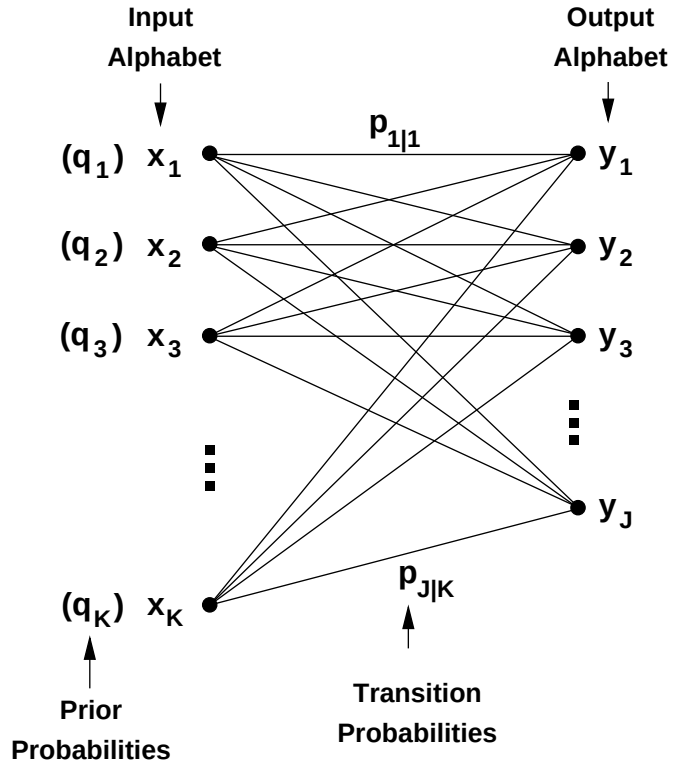


Figure 20: General discrete memoryless channel definition.

The capacity of a DMC is the maximum average mutual information between the input and output, where the maximum is over the prior probabilities,

$$C = \max_{q} I(X;Y) = \max_{q} I(Y;X), \tag{1}$$

where

$$I(X;Y) = E_{XY}\left[\log\left(\frac{P_{X|Y}(x_k|y_j)}{P_X(x_k)}\right)\right]. \tag{2}$$

For this relatively simple channel, the average mutual information is easy to compute and we obtain

$$C = \max_{\boldsymbol{q}} \sum_{j,k} p_{j|k}\, q_k \log\left(\frac{p_{j|k}}{\sum_{l=1}^{K} p_{j|l} q_l}\right) \tag{3}$$

A.2 Finite-State Channels

Here we have a discrete channel with a form of memory. The channel can take on one of A states in each channel-use period. The time-varying channel gives rise to the notion of computing mutual information over the first N channel uses, finding the capacity by maximizing this quantity, and finally letting the number of channel uses increase without bound. The maximization is generally over the prior probabilities for *each* of the channel uses. For special channels, such as decomposable channels in which a state may be reached from another state but may never leave, the priors may indeed need to be different over the different channel-use periods in order to maximize mutual information.

The channel here is modeled as a first-order Markov process. For N channel uses we have the input sequence $\mathsf{x}_1, \ldots, \mathsf{x}_N$ and the output sequence $\mathsf{y}_1, \ldots, \mathsf{y}_N$. Let the vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ denote the input and output sequences, respectively. For each possible value $\boldsymbol{u}$ of $\boldsymbol{x}$ we have a probability measure $Q_N(\boldsymbol{u})$. The probability measure on the output sequence is connected to the prior probabilities and the probability structure of the channel. Let the ith channel state be represented by the random variable S_i which can take on one of A states $s_1, s_2, \ldots, s_A$. By exploiting the Markov channel structure, we can form the conditional output probability given by

$$P_N(\boldsymbol{y}, S_N|\boldsymbol{x}, S_0) = \sum_{S_{N-1}} P(\mathsf{y}_N, S_N|\mathsf{x}_N, S_{N-1}) P_{N-1}(\mathsf{y}_{N-1}, S_{N-1}|\mathsf{x}_{N-1}, S_0),$$

and

$$P_N(\boldsymbol{y}|\boldsymbol{x}, S_0) = \sum_{S_N} P_N(\boldsymbol{y}, S_N|\boldsymbol{x}, S_0).$$

This type of conditional density is necessary since the channel state is allowed to depend on the previous inputs, as required, for example, for simple modeling of inter-symbol interference.

The mutual information—conditional on S_0—can now be determined using Q_N and the conditional output density. Two types of capacity are defined corresponding to the best and worst case values of S_0.

$$\overline{C} = \lim_{N\to\infty} \overline{C}_N, \tag{4}$$

where

$$\overline{C}_N = \max_{\boldsymbol{Q}_N} \max_{S_0} I_Q(X^N; Y^N|S_0),$$

$$I_Q(X^N; Y^N|S_0) = \sum_{\boldsymbol{x},\boldsymbol{y}} Q_N(\boldsymbol{x}) P_N(\boldsymbol{y}|\boldsymbol{x}, S_0) \log\left[\frac{P_N(\boldsymbol{y}|\boldsymbol{x}, S_0)}{\sum_{\boldsymbol{x}'} Q_N(\boldsymbol{x}') P_N(\boldsymbol{y}|\boldsymbol{x}', S_0)}\right],$$

and $\boldsymbol{Q}_N$ is the collection of all the probabilities $Q_N(\boldsymbol{x})$. For the alternate capacity $\underline{C}_N$, replace the maximum over S_0 with a minimum.

A.3 Discrete-Time Memoryless Channels

Here the channel input and output are continuous random variables so that the alphabets can be infinite. The input letters are still applied in succession so that we retain the discrete-time nature of the channel. The basic analysis concept for such channels is to choose a finite subset of the input alphabet and to partition the output space into a finite collection of subsets. Let the input alphabet be $x_1, \ldots, x_K$ with prior probabilities $q_1(x_1), \ldots, q_K(x_K)$, and partition the output space into J mutually exclusive events that exhaust the space. Let these events be denoted by $y_1, \ldots, y_J$. The average mutual information follows as

$$I(X;Y) = \sum_{j,k} q_k(x_k) P_{Y|X}(y_j|x_k) \log \left[\frac{P_{Y|X}(y_j|x_k)}{\sum_{l=1}^{K} q_l(x_l) P_{Y|X}(y_j|x_l)} \right], \tag{5}$$

and the capacity is

$$C = \sum I(X;Y), \tag{6}$$

where the supremum is over all finite selections of the x_k, all priors $q_k(x_k)$, and all output-space partitions y_j. A difficulty with this analysis is that the input letters are not contained in amplitude and therefore any particular noise level (resulting in the transition probabilities for the channel) can be overcome by simply choosing an input alphabet with very large elements. Therefore, an input constraint is often imposed on the input alphabet. Let the constraint function be a real-valued function $f(\cdot)$. Then the input-constrained capacity is given by (5) and (6) with the supremum over all partitions of the output space, all finite selections of the x_k, and all priors q_k such that

$$\sum_{k=1}^{K} q_k(x_k) f(x_k) \leq \mathcal{E}.$$

For example, $f(x) = x^2$ represents an average energy constraint, and $f(x) = |x|$ represents an amplitude constraint.

A.3.1 Discrete-Time Memoryless AWGN Channel

A special case of the general discrete-time memoryless channel is the discrete-time memoryless channel with additive white Gaussian noise (AWGN). Let the channel noise have zero mean and variance σ^2 and impose an average-energy constraint

$$\sum_{k=1}^{K} q_k(x_k) x_k^2 \leq \mathcal{E}.$$

Then the capacity is given by

$$C = \frac{1}{2} \log \left(1 + \frac{\mathcal{E}}{\sigma^2} \right). \tag{7}$$

A.4 The Waveform Channel

For these channels, the input and output are functions of time, typically constrained to the class of L_2 functions on the interval $[0, T]$ (square-integrable functions). To make use of the previously described discrete-alphabet discrete-time machinery, and to simplify the probabilistic analysis, we represent each function as a (possibly infinite) expansion onto a complete orthonormal set of functions. The set of coefficients for a function is then used to specify the function, which allows reasonable and tractable definitions of mutual information and capacity for the waveform channel.

Let $x(t)$ and $y(t)$ denote the input and output channel waveforms, and $\{\phi_n(t)\}$ denote the orthonormal set. Then the expansion coefficients for $x(t)$ are

$$x_n = \int_{-\infty}^{\infty} x(t)\phi_n^*(t)\,dt = \int_0^T x(t)\phi_n^*(t)\,dt,$$

where

$$x(t) = \sum_{n=1}^{\infty} x_n \phi_n(t),$$

and similarly for y_n. Furthermore, let $\boldsymbol{x} = [x_1, \ldots, x_N]$, $\boldsymbol{y} = [y_1, \ldots, y_N]$. Suppose that the conditional probability densities $p_N(\boldsymbol{y}|\boldsymbol{x})$ exist for all finite N. Then these probabilities play the role of channel transition probabilities. Let the prior probability density be denoted by $q_N(\boldsymbol{x})$. Then the mutual information between channel input and output is given by

$$I_T(x(t); y(t)) = \underset{N\to\infty}{\text{l. i. m.}}\, I(\boldsymbol{x}; \boldsymbol{y}),$$

where

$$I(\boldsymbol{x}; \boldsymbol{y}) = \log\left[\frac{p_N(\boldsymbol{y}|\boldsymbol{x})}{\int_{\boldsymbol{x}_1} q_N(\boldsymbol{x}_1 p_N(\boldsymbol{y}|\boldsymbol{x}_1)\,d\boldsymbol{x}_1}\right].$$

The average mutual information is the limiting version of the expected value of the mutual information,

$$I_T(X(t); Y(t)) = \lim_{N\to\infty} E\left[I(\boldsymbol{x}; \boldsymbol{y})\right] = \lim_{N\to\infty} I(\boldsymbol{X}; \boldsymbol{Y}).$$

The capacity is then defined as

$$C = \lim_{T\to\infty} \frac{1}{T} \sup I(\boldsymbol{X}; \boldsymbol{Y}), \tag{8}$$

where the supremum is over all prior density functions consistent with any channel-input constraints.

A.4.1 AWGN Waveform Channel

Let the output of a waveform channel be the sum of the input and WGN with spectral density $N_0/2$. Apply an input-power constraint of S, and let the input be duration limited to an interval of length T and have approximate bandwidth W. Then the capacity per unit time is given by

$$C = W \log\left(1 + \frac{S}{WN_0}\right) \quad \text{bits/sec}, \tag{9}$$

which is Shannon's most famous channel-capacity result.

A.5 Discussion

Presumably the waveform-channel capacity should never be less than the DMC or discrete-time memoryless channel capacities when they are compared properly by constraining the bandwidth and symbol/waveform duration. This conclusion is due to the lack of imposed system structure of the waveform channel with respect to the other channel types; the waveform channel can always be *used* as a discrete-time channel.

For a DMC with K letters in its input alphabet, the capacity for a channel use each T seconds (assuredly achieved at infinite SNR) is simply

$$C_{DMC} = \frac{\log_2 K}{T} \quad \text{bits/sec.}$$

For example, for the BSC, $K = 2$ and

$$C_{BSC} = \frac{\log_2(2)}{T} = \frac{1}{T} \quad \text{bits/sec.}$$

On the other hand, for the waveform channel and any SNR, we have

$$C_{WC} = W \log_2 \left(1 + \frac{S}{WN_0}\right) \quad \text{bits/sec.}$$

Constraining the bandwidth used for the DMC to be about $1/T = W$, we have

$$C_{DMC} = W \log_2 K \quad \text{bits/sec.}$$

Clearly, the waveform channel capacity can be indefinitely increased by increasing the SNR, whereas the DMC capacity cannot. Even for a fixed SNR, the waveform channel capacity can be much larger than that for a given DMC.

Canonical Coordinates for Transform Coding of Random Sources from Noisy Observations

EDICS 3-QUAN

Peter J. Schreier (corresponding author)
Dept. of Electrical and Computer Engineering
University of Colorado
Boulder, CO 80309 – 0425
ph (303) 492-2759 — fax (303) 492-2758
e-mail: pjs@dsp.colorado.edu

Louis L. Scharf, *Fellow, IEEE*
Dept's of Electrical and Computer Engineering and Statistics
Colorado State University
Ft. Collins, CO 80523
ph (970) 491-2979 — fax (970) 491-2249
e-mail: scharf@engr.colostate.edu

This work was supported by the DARPA ISP program under contract AFRL F33615-02-C-1198 and the 2001 NSF ITR Initiative under contract CCR0112573.

May 27, 2003

Abstract

Historically, transform coding of noisy sources has been performed by first estimating the message and then quantizing this estimate. We show that much insight can be gained by recognizing that it is also optimum to first transform the noisy observations into canonical coordinates, quantize, apply a Wiener filter in this coordinate system, and then transform the result back to the original coordinates. Canonical coordinates are uncorrelated, and quantization and Wiener filtering are applied to each component independently. Optimality of this approach can be proved assuming additive white quantization noise. Half canonical coordinates minimize the mean-squared error by minimizing the trace of the error covariance matrix and full canonical coordinates maximize information rate by minimizing the determinant of the error covariance matrix. We also demonstrate in this paper that majorization is the fundamental principle underlying proofs of optimal transform coding, sometimes in a very direct, sometimes in a more indirect way.

Keywords

Transform Coding, Quantization, Canonical Coordinates, Rank Reduction, Majorization

I. INTRODUCTION

In this paper we are interested in the following problem: Given a finite bit-budget of B bits, how can we most efficiently represent the information that a *noisy* observation $\mathbf{y} \in \mathbb{R}^n$ contains about a *random* message $\mathbf{x} \in \mathbb{R}^m$? The apparatus that we will have at our disposal is depicted in Fig. 1. First, the observation $\mathbf{y}$ is passed through a linear transformation $\mathbf{A} \in \mathbb{R}^{m\times n}$, which we call the *coder*. The output of the coder is $\mathbf{u} = \mathbf{A}\mathbf{y}$, which is subsequently processed by a *scalar quantizer*. That is, each component of $\mathbf{u}$ is independently quantized. The quantizer output $\hat{\mathbf{u}}$ is supposed to be an efficient representation of the message $\mathbf{x}$, not the measurement $\mathbf{y}$. To produce an estimate $\hat{\mathbf{x}} = \mathbf{B}\hat{\mathbf{u}}$ of the message $\mathbf{x}$, $\hat{\mathbf{u}}$ is linearly transformed by the *decoder* $\mathbf{B} \in \mathbb{R}^{m\times m}$. Without loss of generality, we suppose that $m \leq n$. Furthermore, we will assume that $\mathbf{x}$ and $\mathbf{y}$ have zero mean and that we have the necessary second-order information available, namely, the covariance matrices of $\mathbf{x}$ and $\mathbf{y}$, denoted by $\mathbf{R}_{xx} = E\mathbf{x}\mathbf{x}^T$ and $\mathbf{R}_{yy} = E\mathbf{y}\mathbf{y}^T$, respectively, and the cross-covariance matrix $\mathbf{R}_{xy} = E\mathbf{x}\mathbf{y}^T$. For an adaptive implementation of our results we would assemble M independent snapshots of $[\mathbf{x}, \mathbf{y}]$ into matrices $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_M]$ and $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_M]$. The covariance matrices $\mathbf{R}_{xx}$, $\mathbf{R}_{xy}$, and $\mathbf{R}_{yy}$ could then be estimated as $M^{-1}\mathbf{X}\mathbf{X}^T$, $M^{-1}\mathbf{X}\mathbf{Y}^T$, and $M^{-1}\mathbf{Y}\mathbf{Y}^T$.

The problem can now be re-formulated as follows: First, how do we choose $\mathbf{A}$ and $\mathbf{B}$, i.e., in what coordinate system should we quantize? Second, how do we distribute the total number of bits B over the components of $\mathbf{u}$ so that $\hat{\mathbf{x}}$ is a good estimate of $\mathbf{x}$? To make precise what we mean by a "good" estimate

	assumptions		Karhunen-Loève Transform		
Paper	AWN	Gauss	$\mathbf{A} = \mathbf{B}^{-1}$	$\mathbf{A}^T = \mathbf{A}^{-1}$	$\mathbf{u}$ uncorr
[1]		■	$\surd$	$\surd$	■
[2, App. I]		■	■	■	$\surd$
[3, Ch. 8.6]	■†	■	■	■	$\surd$
[4, App.]	■		■	$\surd^{\text{wlog}}$	$\surd$
This paper	■		$\surd$	$\surd^{\text{wlog}}$	$\surd$

TABLE I

A SHORT HISTORY OF TRANSFORM CODING – WHAT HAS BEEN ASSUMED (■) AND WHAT HAS BEEN PROVED ($\surd$); $\surd^{\text{wlog}}$ = PROVED THAT IT IS POSSIBLE TO CHOOSE $\mathbf{A}$ AS ORTHOGONAL WITHOUT LOSS OF GENERALITY; ■† = ONLY HIGH-RESOLUTION ASSUMPTION IS USED

we will employ two different performance measures: $E = \operatorname{tr} \mathbf{R}_{ee} = \operatorname{tr} E\mathbf{e}\mathbf{e}^T = E\|\mathbf{x} - \hat{\mathbf{x}}\|^2$, which is the mean squared error (MSE), and $V = \det \mathbf{R}_{ee}$, which measures the volume of the error covariance ellipsoid and thus information rate in the Gaussian case. For simplicity, let us refer to the problems where we try to minimize E and V as the *min-trace* and *min-det* problems, respectively.

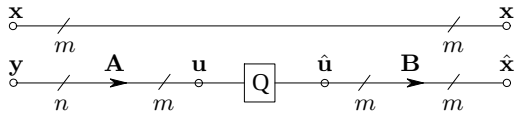


Fig. 1. Transform Coder

A. Historical Overview

Noise-free transform coding: The arrangement in Fig. 1 is commonly known as a transform coder. Most transform coders considered in the literature work on a noiseless measurement $\mathbf{y} = \mathbf{x}$. Given the eigenvalue decomposition $\mathbf{R}_{xx} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, the orthogonal matrix $\mathbf{U}$ is often referred to as the Karhunen-Loève Transform (KLT) corresponding to $\mathbf{x}$. A common claim is the following: "The KLT is optimum for noise-free transform coding, meaning that the choice $\mathbf{A} = \mathbf{U}^T$ and $\mathbf{B} = \mathbf{U}$ (together with a suitable bit assignment strategy) minimizes the mean squared error E."

There are, however, important caveats regarding this claim. They concern the underlying assumptions that have been made in order to prove it. Table I gives a short history of results for noise-free transform coding. Proving optimality of the KLT really means establishing three properties: $\mathbf{A} = \mathbf{B}^{-1}$, $\mathbf{A}$ is

orthogonal, and the quantizer input $\mathbf{u}$ is uncorrelated. However, as can be inferred from Table I, with the exception of this paper, one or two of the KLT's properties have always been *assumed* rather than actually *proved*.

The most important classification of a proof is according to whether or not it utilizes the high-resolution assumption. *High-resolution* means that the bit-budget B is asymptotically large and fine quantizers are employed. If some additional smoothness constraints are made, then quantization noise can be modeled as additive white noise, which is independent of the input signal [5]. This leads to the additive white noise (AWN) model for quantization [6], which we will discuss in Section II-A. Thus, the AWN model implies high-resolution, but not vice versa. Without the use of the AWN model, restrictive assumptions must be made as in proofs [1], [2], [3]. This should come as no surprise. Since a quantizer is an inherently *non-linear* device, we should not expect the solution to the min-trace problem to be the KLT, a result from *linear* algebra, *unless* we have linearized the quantizer through the use of the AWN model, or we have made the problem statement so specific that we basically *force* the solution to be the KLT. In the paper that introduced transform coding, Huang and Schultheiss [1] *assumed* uncorrelated quantizer input and then *proved* that, in order to minimize E, $\mathbf{A}$ must be chosen as $\mathbf{B}^{-1}$ and $\mathbf{A}$ must be orthogonal. Intuition might suggest that requiring uncorrelated quantizer inputs is the right thing to do. Strictly speaking, however, [1] does not prove the optimality of the KLT because its central property is anticipated. Requiring $\mathbf{A} = \mathbf{B}^{-1}$, as [2], [3], [4] do, is more reasonable, because this guarantees $E \to 0$ as $B \to \infty$. The additional assumption of *orthogonal* $\mathbf{A}$ in [2], [3], however, is again restrictive, and we will show in this paper that for performance measures other than MSE orthogonal transforms can be outperformed by non-orthogonal transforms.

We should also mention that without the use of the AWN model, optimality of the KLT can only be proved for Gaussian input. In fact, Zeger [7] has recently shown that, even in the high-resolution case, KLTs can be strictly sub-optimum for transform coding if the input data is non-Gaussian. Thus, in the table, the Gaussian assumption can only be dropped when the AWN model is employed.

In short, only a linearized version of this problem, which uses the AWN model, yields the KLT as the general solution. Otherwise, if the optimality of the KLT can be shown for particular assumptions, this is more an indication that the problem was cleverly posed rather than evidence that the KLT is indeed optimum for transform coding in general.

Transform coding of noisy sources: If the observations are not equal to the message, then it has been shown in [8], [9], that for the min-trace problem it is optimum to first find the MMSE estimate of the

message and then quantize this estimate. This result has been extended in [10] to more general performance measures, which include weighted MSE, and it has been shown in [11] that a particular weighting produces a solution to the min-det problem. This establishes that the min-det problem can also be decomposed into estimator and quantizer. However, as we will demonstrate in this paper, it is not necessary and not necessarily advantageous to estimate first and then quantize.

B. Contribution of This Paper

In this paper, we are concerned with transform coding of random sources from noisy observations. We extend known results in the following ways:

- We show that a possible solution to the min-trace and the min-det problem is to first transform the noisy observations into a half canonical [12, p. 330] or full canonical coordinate system [13] – [17], respectively, quantize, Wiener filter in this coordinate system, and then transform the result back to the original coordinates. Canonical coordinates are uncorrelated, which means quantization and Wiener filtering are applied to each component independently. This extends [8], [9], [10] in that it provides a *concrete* coordinate system for quantization. Moreover, our results show that transform coders have many different implementations: for example, there are implementations where quantization precedes estimation, and vice versa.
- We generalize Table I to the noisy case, giving a proof that invokes the AWN model for quantization, but does not make additional assumptions regarding the transformations $\mathbf{A}$ and $\mathbf{B}$. Moreover, previous proofs in Table I only consider the min-trace problem, but we solve the min-det problem as well.
- We demonstrate that majorization is the fundamental principle underlying proofs of optimal transform coding, sometimes in a very direct, sometimes in a more indirect way.
- We establish an important connection between quantization and rank reduction: It has been shown in [11], [12, p. 330] that we should use half or full canonical coordinates for rank reduction, as well. From a quantization point of view, rank reduction means assigning infinitely many bits to a number of components and zero bits to the remaining components, which is sometimes also called *zonal sampling*. Together with our results, this means that we can *first* choose a coordinate system and *then* decide how many bits to spend on how many components.

Our program for this paper is as follows: In Section II-A we give a short introduction to quantization using the AWN model, and in Section II-B we present a concise overview of some majorization results. Sections III and IV prove that, under the min-trace and min-det criterion, the right coordinate systems in which to perform quantization are half and full canonical coordinates, respectively. Finally, Section V

looks at several different implementations of rank reduction and quantization in canonical coordinates. Each realization has its benefits and brings its own insights.

II. PREREQUISITES

A. Quantization

A quantizer can always be modeled as an additive noise source, meaning that the quantizer output $\hat{\mathbf{u}} = \mathbf{u} + \mathbf{q}$ is equal to the quantizer input $\mathbf{u}$ plus quantization noise $\mathbf{q}$. However, an MMSE Lloyd–Max quantizer [3, Ch. 6.2] only guarantees that $E\mathbf{q} = \mathbf{0}$, $E\hat{u}_i q_i = 0$, and $E\hat{u}_i u_i = E\hat{u}_i^2$. In general, cross-terms such as $Eq_i q_j$ and $Eu_i q_j$ are non-zero and given by complicated expressions. In order to make the quantization problem analytically tractable, it is common to employ the additive white noise (AWN) model. It is based on the high-resolution assumption (fine quantizers with large number of bits) and additional smoothness constraints [5], [6]. If we let b_i denote the number of bits for quantizing component u_i, and $\sigma_{u_i}^2 = Eu_i^2$ the variance of u_i, then the main assumptions of the AWN model may be summarized as follows:

$$E\mathbf{q}\mathbf{q}^T = \operatorname{diag}(\sigma_{q_1}^2, ..., \sigma_{q_m}^2) \tag{1}$$

$$\sigma_{q_i}^2 = Eq_i^2 = c\sigma_{u_i}^2 2^{-2b_i}, \quad i = 1, ..., m \tag{2}$$

$$E\mathbf{u}\mathbf{q}^T = \mathbf{0} \tag{3}$$

The constant c is dependent on the distribution of u_i. If u_i is zero-mean Gaussian, then $c = \sqrt{3}\pi/2$ [3, Ch. 8.2]. The advantage of the AWN model is that a quantizer is modeled as an additive *white* noise source that is *uncorrelated* with the input signal. Thus, the quantizer — an inherently non-linear device — has been linearized.

Property (2) is a consequence of the high-resolution assumption. Without invoking the high-resolution assumption, we can model the variance of q_i more generally as

$$Eq_i^2 = \sigma_{u_i}^2 f(b_i) \tag{4}$$

in the Gaussian case. Here, $f(b_i)$ is a non-increasing function of the number of bits spent on component u_i. Clearly, we expect to get better performance by increasing b_i.

B. Majorization and Schur-Convex Functions

As suggested in the Introduction, majorization plays a central role in quantization. In this section we introduce the concept.

Definition 1 (Majorization) [18, p. 7] A real $n \times 1$ vector $\mathbf{x}$ is said to be *majorized* by a real $n \times 1$ vector $\mathbf{y}$, written as $\mathbf{x} \prec \mathbf{y}$, if

$$\sum_{i=1}^{k} x_{[i]} \leq \sum_{i=1}^{k} y_{[i]}, \quad k = 1, ..., n-1 \tag{5}$$

$$\sum_{i=1}^{n} x_{[i]} = \sum_{i=1}^{n} y_{[i]}. \tag{6}$$

where $[\cdot]$ is a permutation operator such that $x_{[1]} \geq \cdots \geq x_{[n]}$.

Intuitively, if $\mathbf{x} \prec \mathbf{y}$, then the components of $\mathbf{x}$ are "less spread out" or "more equal" than the components of $\mathbf{y}$. Note that majorization is sometimes also defined with respect to a permutation operator that arranges the components of $\mathbf{x}$ in *increasing* order [19, Def. 4.3.24], $x_{[1]} \leq \cdots \leq x_{[n]}$.

The idea of majorization becomes most powerful when it is combined with the concept of Schur-convexity. Functions that are Schur-convex preserve the partial ordering of majorization:

Definition 2 (Schur-convex function) [18, Def. 3.A.1] A real-valued function g defined on a set $D \subset \mathbb{R}^n$ is said to be *Schur-convex on* D if $\mathbf{x} \prec \mathbf{y}$ on D implies that $g(\mathbf{x}) \leq g(\mathbf{y})$. Similarly, a function is called *Schur-concave on* D if $\mathbf{x} \prec \mathbf{y}$ on D implies that $g(\mathbf{x}) \geq g(\mathbf{y})$.

Schur-convex functions are necessarily symmetric when they are defined on $\mathbb{R}^n$. However, functions that are not symmetric on $\mathbb{R}^n$ can still be Schur-convex on the set of ordered n-tuples, which we define as

$$\mathcal{D}_n = \{(x_1, ..., x_n) : x_1 \geq \cdots \geq x_n\}. \tag{7}$$

To prove that a function is Schur-convex or Schur-concave, there are a number of results, which can be found in [18, Ch. 3]. We will use the following proposition:

Proposition 1: [18, 3.H.2] Let $g(\mathbf{x}) = \sum_{i=1}^{n} h_i(x_i)$, $\mathbf{x} \in \mathcal{D}_n$, where each $h_i : \mathbb{R} \to \mathbb{R}$ is differentiable. Then g is Schur-convex on $\mathcal{D}_n$ if and only if

$$h_i'(a) \geq h_{i+1}'(b), \quad a \geq b, i = 1, ..., n-1 \tag{8}$$

where $h_i'(a)$ denotes the first derivative of h_i evaluated at a. Analogously, if $h_i'(a) \leq h_{i+1}'(b)$ whenever $a \geq b$, $i = 1, ..., n-1$, then g is Schur-concave.

A classical result of majorization is that if $\mathbf{H}$ is an $n \times n$ Hermitian matrix with diagonal elements $\mathbf{diag}(\mathbf{H}) = (H_{11}, ..., H_{nn})^T$ and eigenvalues $\mathbf{ev}(\mathbf{H}) = (\lambda_1, ..., \lambda_n)^T$, then [18, Ch. 9.B]

$$\mathbf{diag}(\mathbf{H}) \prec \mathbf{ev}(\mathbf{H}). \tag{9}$$

Since $g(\mathbf{x}) = \prod_{i=1}^{n} x_i$ is a Schur-concave function [18, 3.F.1], this immediately proves Hadamard's inequality:

$$\prod_{i=1}^{n} H_{ii} \geq \det \mathbf{H}. \tag{10}$$

Many other inequalities such as the arithmetic mean/geometric mean (AM/GM) inequality or Minkowski's inequality can be viewed as consequences of majorization, as well [18].

III. Half Canonical Coordinates Solve the min-trace or MMSE Problem

In this section, we show that for the min-trace problem the right coordinate system for quantization is the system of *half canonical coordinates* [12, p. 330]. We will provide two proofs: one that is based on the AWN model, and one that starts with more restrictive assumptions, but does not use the high-resolution assumption. We will demonstrate how majorization is the underlying principle for both proofs.

We refer to the notation introduced in Fig. 1. The starting point for both proofs is the error vector $\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}}$, which, without any additional assumptions, is given by

$$\mathbf{e} = (\mathbf{BAy} - \mathbf{x}) + \mathbf{Bq}. \tag{11}$$

A. Additive White Noise Model

If we invoke the AWN model, then $E\mathbf{xq}^T = \mathbf{0}$ and $E\mathbf{yq}^T = \mathbf{0}$, and the error covariance matrix $\mathbf{R}_{ee} = E\mathbf{ee}^T$ becomes

$$\mathbf{R}_{ee} = E\left[(\mathbf{BAy} - \mathbf{x})(\mathbf{BAy} - \mathbf{x})^T\right] + \mathbf{BR}_{qq}\mathbf{B}^T, \tag{12}$$

which can be expressed as the sum of three positive semi-definite terms:

$$\mathbf{R}_{ee} = \mathbf{Q} + (\mathbf{W} - \mathbf{BA})\mathbf{R}_{yy}(\mathbf{W} - \mathbf{BA})^T + \mathbf{BR}_{qq}\mathbf{B}^T \tag{13}$$

In this equation, $\mathbf{W} = \mathbf{R}_{xy}\mathbf{R}_{yy}^{-1}$ is the Wiener filter and $\mathbf{Q} = \mathbf{R}_{xx} - \mathbf{R}_{xy}\mathbf{R}_{yy}^{-1}\mathbf{R}_{xy}^T$ is its filtering error covariance matrix. It is clear that we can make the middle term in (13) zero if we choose $\mathbf{BA} = \mathbf{W}$, and we will assume this optimum choice in what follows. Thus the infinite precision quantizer is a Wiener filter with error covariance $\mathbf{Q}$. Since $\mathbf{Q}$ does not depend on how we select $\mathbf{A}$ and $\mathbf{B}$, minimizing $\mathrm{tr}\,\mathbf{R}_{ee}$ amounts to minimizing $\mathrm{tr}\,\mathbf{BR}_{qq}\mathbf{B}^T$. This can be achieved with a variation on the proof for the noiseless

case in [4, Appendix]. Denote the i-th column of $\mathbf{B}$ by $\mathbf{b}_i$. We then have from (1) and (2)

$$\operatorname{tr} \mathbf{B}\mathbf{R}_{qq}\mathbf{B}^T = \sum_{i=1}^{m} c2^{-2b_i}\sigma_{u_i}^2 \|\mathbf{b}_i\|^2 \tag{14}$$

$$\geq cm2^{-2b}\left[\prod_{i=1}^{m}\sigma_{u_i}^2\|\mathbf{b}_i\|^2\right]^{1/m}. \tag{15}$$

The inequality is an AM/GM inequality, where we have defined

$$b = \left(\prod_{i=1}^{m} b_i\right)^{1/m} \tag{16}$$

as the geometric mean of the bit rates b_i. Since $\sigma_{u_i}^2 = (\mathbf{A}\mathbf{R}_{yy}\mathbf{A}^T)_{ii}$, Hadamard's inequality yields

$$\prod_{i=1}^{m}\sigma_{u_i}^2 \geq \det(\mathbf{A}\mathbf{R}_{yy}\mathbf{A}^T). \tag{17}$$

Using this inequality and the fact that $\det \mathbf{B}\mathbf{B}^T = (\det \mathbf{B})^2$ in (15) we obtain a new lower bound

$$\operatorname{tr} \mathbf{B}\mathbf{R}_{qq}\mathbf{B}^T \geq cm2^{-2b}[\det(\mathbf{A}\mathbf{R}_{yy}\mathbf{A}^T)\det(\mathbf{B}\mathbf{B}^T)]^{1/m}\left[\frac{\prod_{i=1}^{m}\|\mathbf{b}_i\|^2}{(\det\mathbf{B})^2}\right]^{1/m}. \tag{18}$$

This expression can in turn be lower bounded by using Hadamard's inequality once more to arrive at

$$\operatorname{tr} \mathbf{B}\mathbf{R}_{qq}\mathbf{B}^T \geq cm2^{-2b}[\det(\mathbf{A}\mathbf{R}_{yy}\mathbf{A}^T)\det(\mathbf{B}\mathbf{B}^T)]^{1/m} \tag{19}$$

$$= cm2^{-2b}[\det(\mathbf{B}\mathbf{A}\mathbf{R}_{yy}\mathbf{A}^T\mathbf{B}^T)]^{1/m} \tag{20}$$

$$= cm2^{-2b}[\det(\mathbf{R}_{xy}\mathbf{R}_{yy}^{-1}\mathbf{R}_{xy}^T)]^{1/m}. \tag{21}$$

This final lower bound can be achieved if the inequalities we have used become equalities. For the Hadamard inequalities this means that

$$\mathbf{A}\mathbf{R}_{yy}\mathbf{A}^T = \mathbf{D}_1 \tag{22}$$

$$\mathbf{B}\mathbf{B}^T = \mathbf{D}_2, \tag{23}$$

where $\mathbf{D}_1$ and $\mathbf{D}_2$ are both $m \times m$ diagonal matrices. The two conditions (22) and (23) determine the *coordinate system* for $\mathbf{u}$. The AM/GM inequality, on the other hand, becomes an equality if

$$c2^{-2b_i}\left(\mathbf{A}\mathbf{R}_{yy}\mathbf{A}^T\right)_{ii}\|\mathbf{b}_i\|^2 = K, \tag{24}$$

where K is independent of i. This determines the *bit assignment* for $\mathbf{u}$.

Let us first talk about the coordinate system. From (22) and (23), and since $\mathbf{B}\mathbf{A} = \mathbf{W}$, we find that $\mathbf{B}^T\mathbf{R}_{xy}\mathbf{R}_{yy}^{-1}\mathbf{R}_{xy}^T\mathbf{B} = \mathbf{D}_2^T\mathbf{D}_1\mathbf{D}_2^T$, which implies that $\mathbf{B}$ must diagonalize $\mathbf{R}_{xy}\mathbf{R}_{yy}^{-1}\mathbf{R}_{xy}^T$. We could thus

choose $\mathbf{B}$ as the orthogonal matrix $\mathbf{U}$ from the eigenvalue decomposition $\mathbf{R}_{xy}\mathbf{R}_{yy}^{-1}\mathbf{R}_{xy}^T = \mathbf{U}(\mathbf{Z}\mathbf{Z}^T)\mathbf{U}^T$, and $\mathbf{A} = \mathbf{U}^T\mathbf{R}_{xy}\mathbf{R}_{yy}^{-1} = \mathbf{U}^T\mathbf{W}$. With this choice, we are first estimating $\mathbf{x}$ by passing $\mathbf{y}$ through a Wiener filter $\mathbf{W}$, and then quantizing the estimate $\tilde{\mathbf{x}} = \mathbf{W}\mathbf{y}$, as in the noise-free case. The noisy quantization problem is thus reduced to a standard quantization problem, as long as we observe that the estimate $\tilde{\mathbf{x}}$ has covariance matrix $\mathbf{R}_{xy}\mathbf{R}_{yy}^{-1}\mathbf{R}_{xy}^T$ rather than $\mathbf{R}_{xx}$. This result connects to a finding in [8] and [9]. There it was demonstrated that

$$\begin{aligned} E &= E\|\mathbf{x} - \hat{\mathbf{x}}\|^2 \\ &= E\|\mathbf{x} - E(\mathbf{x}|\mathbf{y})\|^2 + E\|E(\mathbf{x}|\mathbf{y}) - \hat{\mathbf{x}}\|^2 \qquad (25) \\ &= E_{\text{filter}} + E_{\text{quantizer}}, \qquad (26) \end{aligned}$$

which shows that for the MSE criterion it is optimum to apply the quantizer to a *conditional mean* estimator of the message based on the observations. The total MSE E is the sum of the infinite precision filtering error and the error of quantizing the conditional mean estimate. Our result differs insofar as we have shown that, using the AWN model for quantization and a transform coding system, it is optimum to first obtain a *linear* MMSE estimate $\tilde{\mathbf{x}} = \mathbf{W}\mathbf{y}$ through the Wiener filter, and then quantize $\tilde{\mathbf{x}}$. Gaussianity is not required for our proof. For jointly Gaussian message and observations our finding coincides with [8], [9], but in general, they are different.

Moreover, our derivation also allows a different interpretation which brings fresh insight. If we start with the singular value decomposition (SVD) [12, p. 330]

$$\mathbf{R}_{xy}\mathbf{R}_{yy}^{-1/2} = \mathbf{U}\mathbf{Z}\mathbf{V}^T = \mathbf{U}\begin{bmatrix}\mathbf{Z}_m & \mathbf{0}_{m\times(n-m)}\end{bmatrix}\begin{bmatrix}\mathbf{V}_m^T \\ \mathbf{V}_0^T\end{bmatrix}, \qquad (27)$$

one can check that taking $\mathbf{A} = \mathbf{Z}\mathbf{V}^T\mathbf{R}_{yy}^{-1/2}$ and $\mathbf{B} = \mathbf{U}$ also satisfies (22) and (23). Thus $\mathbf{u} = \mathbf{Z}\mathbf{V}^T\mathbf{R}_{yy}^{-1/2}\mathbf{y}$, with covariance $\mathbf{R}_{uu} = \mathbf{Z}_m\mathbf{Z}_m^T$, is quantized for $\hat{\mathbf{u}}$, and $\mathbf{x}$ is then estimated as $\hat{\mathbf{x}} = \mathbf{U}\hat{\mathbf{u}}$. The diagonal elements of $\mathbf{Z}_m$ are the *half canonical correlations* z_i between $\mathbf{x}$ and $\mathbf{y}$. We can now express the MMSE in terms of z_i as

$$\min_{\mathbf{A},\mathbf{B}} E = \left[\operatorname{tr}\mathbf{R}_{xx} - \sum_{i=1}^{m} z_i^2\right] + c\prod_{i=1}^{m} 2^{-2b_i}z_i^2. \qquad (28)$$

The first term in (28) accounts for the infinite-precision filtering error and the second term for the error due to quantization.

The MSE E will be minimized if bits are assigned according to (24), which says that

$$c2^{-2b_i}z_i^2 = K, \qquad (29)$$

subject to $B = \sum_{i=1}^{m} b_i$. The solution to the bit assignment problem parallels the one for standard transform coding if we observe that the variance of u_i is z_i^2. Components u_i with greater squared half canonical correlation z_i^2 will be assigned more bits and according to [3, Ch. 8.3] we have

$$b_i = \frac{B}{m} + \tfrac{1}{2}\log_2 \frac{z_i^2}{\left(\prod_{j=1}^{m} z_j^2\right)^{1/m}}. \tag{30}$$

Note that (22) and (23) allow the transformations $\mathbf{A}$ and $\mathbf{B}$ to be scaled by a non-singular diagonal matrix. If $\mathbf{A}$ is replaced by $\mathbf{D}^{-1}\mathbf{A}$ and $\mathbf{B}$ by $\mathbf{BD}$, then the quantizer input $\mathbf{u}$ will still be uncorrelated and according to (24) the optimum bit assignment is left unchanged. For instance, we could choose $\mathbf{A} = \mathbf{V}_m^T \mathbf{R}_{yy}^{-1/2}$ and $\mathbf{B} = \mathbf{U}\mathbf{Z}_m$. With this choice, the transformation $\mathbf{u} = \mathbf{A}\mathbf{y}$ takes $\mathbf{y}$ into a half canonical coordinate system, where the white, unit-variance, half canonical coordinates $\mathbf{u}$ are quantized. The transformation $\mathbf{B}$ applies a diagonal Wiener filter $\mathbf{Z}_m$ in canonical coordinates to the quantizer output $\hat{\mathbf{u}}$, and transforms the filter output back into the original coordinate system. Note that in this implementation quantization precedes estimation. In Section V we further explore how canonical correlations illuminate quantization. We also look at different realizations of the min-trace quantizer.

B. No High-Resolution Assumption

If we do not use the high-resolution assumption, meaning that the AWN model cannot be employed, either, we should not expect a half canonical coordinate system to be optimum in all generality. However, for the min-trace problem, we can prove the following: Suppose that $\mathbf{x}$ and $\mathbf{y}$ are jointly Gaussian and $\mathbf{A} = \mathbf{U}^T\mathbf{W}$, $\mathbf{B} = \mathbf{U}$, where $\mathbf{U}$ is orthogonal. Furthermore, we model the variance of the quantization noise as in (4). Then given any particular bit assignment vector $\mathbf{b} = (b_1, ..., b_N)^T$, the MSE E is minimized if $\mathbf{A}\mathbf{y}$ produces an uncorrelated $\mathbf{u}$, i.e., $\mathbf{U}$ diagonalizes $\mathbf{R}_{xy}\mathbf{R}_{yy}^{-1}\mathbf{R}_{xy}^T$. While the problem is very restrictive, the proof is nevertheless interesting because it makes obvious the role that majorization plays in quantization, as we now demonstrate.

First, notice that since message and observation are jointly Gaussian and $\mathbf{A} = \mathbf{U}^T\mathbf{W}$, we can use the result from [8], [9], detailed in (25), and only concern ourselves with minimizing

$$E_{\text{quantizer}} = E\|\mathbf{W}\mathbf{y} - \hat{\mathbf{x}}\|^2 = E\|\mathbf{U}(\mathbf{u} - \hat{\mathbf{u}})\|^2 = E\|\mathbf{u} - \hat{\mathbf{u}}\|^2 = \sum_{i=1}^{m} \sigma_{u_i}^2 f(b_i). \tag{31}$$

This means we must simply demonstrate that a KLT is optimum for quantizing $\tilde{\mathbf{x}} = \mathbf{W}\mathbf{y}$, which has covariance $\mathbf{R}_{xy}\mathbf{R}_{yy}^{-1}\mathbf{R}_{xy}^T$.

The following generalizes [2, Proof 2, App. I]. Let us first re-order the components of $\mathbf{u}$ such that $\sigma_{u_i}^2 \geq \sigma_{u_{i+1}}^2$, $i = 1, ..., m-1$. Then note that minimization of (31) requires that $b_i \geq b_{i+1}$, $i =$

$1, ..., m-1$, because $f(b_i)$ is a non-increasing function. This conforms with intuition: We should clearly assign components with greater variance more bits. With these assumptions, $\boldsymbol{\sigma} = (\sigma_{u_1}^2, ..., \sigma_{u_m}^2)^T \in \mathcal{D}_m$ and $\mathbf{b} \in \mathcal{D}_m$ are both members of the set of ordered m-tuples $\mathcal{D}_m$.

The quantization error $E_{\text{quantizer}}$ is a function of $\boldsymbol{\sigma}$. It follows from the majorization result

$$\boldsymbol{\sigma} = \mathbf{diag}(\mathbf{R}_{uu}) \prec \mathbf{ev}(\mathbf{R}_{uu}) = \mathbf{ev}(\mathbf{R}_{xy}\mathbf{R}_{yy}^{-1}\mathbf{R}_{xy}^T) \tag{32}$$

that in order to minimize any Schur-concave function of $\boldsymbol{\sigma} \in \mathcal{D}_m$, $\mathbf{U}$ must diagonalize $\mathbf{R}_{xy}\mathbf{R}_{yy}^{-1}\mathbf{R}_{xy}^T$. Thus, if we can show that our performance measure is Schur-concave on $\mathcal{D}_m$, we have proved optimality of the KLT. First notice that since f is non-increasing and $b_i \geq b_{i+1}$, we have $f(b_i) \leq f(b_{i+1})$. It then follows immediately from Prop. 1 that $E_{\text{quantizer}}$ is Schur-concave.

Our proof is more general than the proof in [2] because it shows that the KLT is optimum for noise-free transform coding for all Schur-concave performance measures, not just MSE. Of course, this statement holds only for the assumptions stated at the beginning of this section. In particular, $\mathbf{U}$ must be orthogonal. The reason this proof can not be extended to non-orthogonal transformations is that $\|\mathbf{U}(\mathbf{u} - \hat{\mathbf{u}})\|^2$ in general depends on the cross-correlations Eq_iq_j, $i \neq j$, unless $\mathbf{U}$ is orthogonal. These cross-correlations are given by complicated expressions, which do not easily admit a solution to the minimization problem $\min E_{\text{quantizer}}$.

C. The Role of Majorization

The proof in the previous section, which does not invoke the high-resolution assumption, makes the role of majorization obvious. If we had complete control over how to distribute $\text{tr}\,\mathbf{R}_{uu}$ over the diagonal elements $\boldsymbol{\sigma}$, then clearly $E_{\text{quantizer}}$ as given by (31) would be minimized by choosing $\sigma_{u_1}^2 = \text{tr}\,\mathbf{R}_{uu}$, $\sigma_{u_i}^2 = 0$, $i = 2, ..., m$, and spending the entire bit-budget B on quantizing component u_1. The question is how close we can come to this rank-1 choice while observing the constraint that $\mathbf{u} = \mathbf{U}\tilde{\mathbf{x}}$. Majorization gives the answer. It is apparent that the bigger the spread in the vector $\boldsymbol{\sigma}$, the smaller MSE will be. Since $\boldsymbol{\sigma} \prec \mathbf{ev}(\mathbf{R}_{xy}\mathbf{R}_{yy}^{-1}\mathbf{R}_{xy}^T)$, the maximum spread in $\boldsymbol{\sigma}$, and therefore minimum MSE, is achieved when $\mathbf{U}$ is a KLT.

Let us now demonstrate how other proofs of the KLT's MSE optimality make use of majorization. Goyal *et al.* in [2, Proof 1, App. I] show, under the same restrictive requirements as the proof in Section III-B, that given any orthogonal transformation $\mathbf{T}$, there exists a KLT $\mathbf{U}$ that yields MSE at most as high as $\mathbf{T}$. They proceed by constructing a series of Jacobi rotations $\{\mathbf{J}_i\}$, which *iteratively* diagonalize $\mathbf{T}E(\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T)\mathbf{T}^T$. Each $\mathbf{J}_i$ makes one off-diagonal element zero, and acts only on two diagonal elements,

increasing one by δ, and decreasing the other one by δ. Since this increases the spread of the diagonal elements, quantizing $\mathbf{J}_{i+1}\mathbf{J}_i \cdots \mathbf{J}_1\mathbf{T}\tilde{\mathbf{x}}$ is better than quantizing $\mathbf{J}_i \cdots \mathbf{J}_1\mathbf{T}\tilde{\mathbf{x}}$. Because $\mathbf{U} = \mathbf{J}_k \cdots \mathbf{J}_2\mathbf{J}_1\mathbf{T}$, this iteratively shows optimality of the KLT. In essence, this construction is a complicated proof of $\mathbf{diag}(\mathbf{R}_{uu}) \prec \mathbf{ev}(\mathbf{R}_{uu})$, as we now show. Each transformation $\mathbf{J}_i$ acts on the diagonal elements of $\mathbf{J}_{i-1} \cdots \mathbf{J}_1\mathbf{T}E(\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T)\mathbf{T}^T\mathbf{J}_1^T \cdots \mathbf{J}_{i-1}^T$ as a so-called T-transformation. A T-transformation has the form $T(\mathbf{z}) = (z_1, ..., z_{k-1}, \alpha z_k + (1-\alpha)z_l, z_{k+1}, ..., z_{l-1}, (1-\alpha)z_k + \alpha z_l, z_{l+1}, ..., z_n)^T$, where $\alpha \in [0, 1]$. Since $\mathbf{ev}(\mathbf{R}_{uu})$ can be derived from $\mathbf{diag}(\mathbf{R}_{uu})$ by successive applications of T-transformations, we have $\mathbf{diag}(\mathbf{R}_{uu}) \prec \mathbf{ev}(\mathbf{R}_{uu})$ [18, Ch. 4].

In a more general setting, we have proved optimality of half canonical coordinates for quantization in Section III-A. The key step is Hadamard's inequality (17). This inequality is a direct consequence of the majorization result $\boldsymbol{\sigma} \prec \mathbf{ev}(\mathbf{R}_{uu})$, as we have already demonstrated in Section II-B. Achieving equality in this inequality requires the largest possible spread among the diagonal elements of $\mathbf{R}_{uu} = \mathbf{A}\mathbf{R}_{yy}\mathbf{A}^T$. This leads to a diagonal matrix $\mathbf{R}_{uu}$ with squared half canonical correlations on its diagonal.

IV. Full Canonical Coordinates Solve the min-det or Maximum Information Rate Problem

In this section, we show that the right coordinate system for $\mathbf{u}$ to solve the min-det problem is the system of *full canonical coordinates* [13] – [17]. The proof will be based on the AWN model. It does not seem possible to extend it to the non-high resolution case, not even under the restrictive assumptions of Section III-B. The reason is that $\det \mathbf{R}_{ee}$ depends on the cross-correlations Eq_iq_j, which are generally non-zero in the absence of high-resolution.

We start the minimization of $V = \det \mathbf{R}_{ee}$ by applying Minkowski's determinant inequality to (13):

$$V \geq \left(\left(\det(\mathbf{Q} + \mathbf{B}\mathbf{R}_{qq}\mathbf{B}^T)\right)^{1/m} + \left(\det\left((\mathbf{W} - \mathbf{B}\mathbf{A})\mathbf{R}_{yy}(\mathbf{W} - \mathbf{B}\mathbf{A})^T\right)\right)^{1/m} \right)^m \tag{33}$$

Since both the term in the left $\det(\cdot)$ and the right $\det(\cdot)$ expression are positive semi-definite, we can minimize V by making the second term zero, i.e., choosing $\mathbf{B}\mathbf{A} = \mathbf{W}$. We will assume this optimum choice in what follows. Using Minkowski's determinant inequality once more yields

$$V \geq \left((\det \mathbf{Q})^{1/m} + \left(\det(\mathbf{B}\mathbf{R}_{qq}\mathbf{B}^T)\right)^{1/m} \right)^m . \tag{34}$$

Since $\mathbf{Q}$ does not depend on the choice of $\mathbf{A}$ or $\mathbf{B}$, $\det(\mathbf{B}\mathbf{R}_{qq}\mathbf{B}^T)$ must be minimized in order to minimize

the bound on V. Similar to the procedure in the previous section we have

$$\det(\mathbf{B}\mathbf{R}_{qq}\mathbf{B}^T) = (c2^{-2b})^m \prod_{i=1}^{m} \left(\mathbf{A}\mathbf{R}_{yy}\mathbf{A}^T\right)_{ii} \det(\mathbf{B}^T\mathbf{B}) \tag{35}$$

$$\geq (c2^{-2b})^m \det(\mathbf{A}\mathbf{R}_{yy}\mathbf{A}^T) \det(\mathbf{B}^T\mathbf{B}) \tag{36}$$

$$= (c2^{-2b})^m \det(\mathbf{R}_{xy}\mathbf{R}_{yy}^{-1}\mathbf{R}_{xy}^T). \tag{37}$$

Inequality (36) is again Hadamard's inequality, which becomes an equality if $\mathbf{A}\mathbf{R}_{yy}\mathbf{A}^T$ is diagonal. Minkowski's inequality (34), on the other hand, becomes an equality if

$$\mathbf{Q} = K \cdot \mathbf{B}\mathbf{R}_{qq}\mathbf{B}^T \tag{38}$$

for some $K \geq 0$. With the knowledge that $\mathbf{A}\mathbf{R}_{yy}\mathbf{A}^T$ must be diagonal for equality, this means

$$\mathbf{Q} = cK \cdot \mathbf{B}[(\mathbf{A}\mathbf{R}_{yy}\mathbf{A}^T)\mathrm{diag}\,(2^{-2b_1}, ..., 2^{-2b_m})]\mathbf{B}^T. \tag{39}$$

The expression in square brackets is a diagonal matrix. Thus, in order to evaluate what (39) implies for $\mathbf{B}$, we would like to factor $\mathbf{Q} = \mathbf{T}\mathbf{D}_Q\mathbf{T}^T$, where $\mathbf{D}_Q$ is diagonal. To this end, we need the SVD of the coherence matrix [17]

$$\mathbf{R}_{xx}^{-1/2}\mathbf{R}_{xy}\mathbf{R}_{yy}^{-T/2} = \mathbf{F}\mathbf{K}\mathbf{G}^T = \mathbf{F}\begin{bmatrix}\mathbf{K}_m & \mathbf{0}_{m\times(n-m)}\end{bmatrix}\begin{bmatrix}\mathbf{G}_m^T \\ \mathbf{G}_0^T\end{bmatrix}. \tag{40}$$

Then we can re-write (39) as

$$\mathbf{Q} = \mathbf{R}_{xx}^{1/2}\mathbf{F}(\mathbf{I} - \mathbf{K}\mathbf{K}^T)\mathbf{F}^T\mathbf{R}_{xx}^{T/2} = cK \cdot \mathbf{B}[(\mathbf{A}\mathbf{R}_{yy}\mathbf{A}^T)\mathrm{diag}\,(2^{-2b_1}, ..., 2^{-2b_m})]\mathbf{B}^T. \tag{41}$$

It is apparent that the following are possible choices for $\mathbf{A}$ and $\mathbf{B}$ such that $\mathbf{B}\mathbf{A} = \mathbf{W}$, $\mathbf{A}\mathbf{R}_{yy}\mathbf{A}^T$ is diagonal, and (41) is satisfied:

$$\mathbf{A} = \mathbf{D}^{-1}\mathbf{K}\mathbf{G}^T\mathbf{R}_{yy}^{-1/2} \tag{42}$$

$$\mathbf{B} = \mathbf{R}_{xx}^{1/2}\mathbf{F}\mathbf{D} \tag{43}$$

Here, $\mathbf{D}$ is any non-singular diagonal matrix. For $\mathbf{D} = \mathbf{K}_m$, the transformation $\mathbf{u} = \mathbf{A}\mathbf{y} = \mathbf{G}_m^T\mathbf{R}_{yy}^{-1/2}\mathbf{y}$ takes $\mathbf{y}$ into the full canonical coordinate system, where the white, unit-variance, full canonical coordinates $\mathbf{u}$ are quantized. The transformation $\mathbf{B} = \mathbf{R}_{xx}^{1/2}\mathbf{F}\mathbf{K}_m$ applies a diagonal Wiener filter $\mathbf{K}_m$ in full canonical coordinates to the quantizer output, and the filter output is transformed back into the original coordinate system with $\mathbf{R}_{xx}^{1/2}\mathbf{F}$. The diagonal Wiener filter $\mathbf{K}_m$ contains the *full canonical correlations* k_i between $\mathbf{x}$ and $\mathbf{y}$ on its diagonal.

It is instructive to express the minimum achievable value of V in terms of k_i:

$$\min_{\mathbf{A},\mathbf{B}} V = \left((\det \mathbf{Q})^{1/m} + c2^{-2b}(\det \mathbf{R}_{xy}\mathbf{R}_{yy}^{-1}\mathbf{R}_{xy}^T)^{1/m}\right)^m \tag{44}$$

$$= \det(\mathbf{R}_{xx}) \cdot \left[\left(\prod_{i=1}^m (1-k_i^2)\right)^{1/m} + c2^{-2b}\left(\prod_{i=1}^m k_i^2\right)^{1/m}\right]^m \tag{45}$$

The first term is the infinite precision filtering error, and the second term is due to quantization [16]. Equation (41) also determines the right bit assignment strategy. We immediately obtain

$$1 - k_i^2 = cK \cdot 2^{-2b_i} k_i^2 \quad i = 1, ..., m. \tag{46}$$

If we define $\gamma_i^2 = k_i^2/(1-k_i^2)$, we must satisfy

$$c2^{-2b_i}\gamma_i^2 = K. \tag{47}$$

This means we have the same solution as for (29) with half canonical correlations z_i^2 replaced by γ_i^2,

$$b_i = \frac{B}{m} + \tfrac{1}{2}\log_2 \frac{\gamma_i^2}{\left(\prod_{j=1}^m \gamma_j^2\right)^{1/m}}. \tag{48}$$

Notice that, just like the min-trace problem, the min-det problem can also be solved by first computing the linear MMSE estimate of $\mathbf{x}$ as $\tilde{\mathbf{x}} = \mathbf{W}\mathbf{y}$ and then quantizing $\tilde{\mathbf{x}}$. To see this, write $\mathbf{A} = \mathbf{K}\mathbf{G}^T\mathbf{R}_{yy}^{-1/2} = \mathbf{F}^T\mathbf{R}_{xx}^{-1/2}\mathbf{R}_{xy}\mathbf{R}_{yy}^{-1} = \mathbf{F}^T\mathbf{R}_{xx}^{-1/2}\mathbf{W}$, and $\mathbf{B} = \mathbf{R}_{xx}^{1/2}\mathbf{F}$. Therefore, this problem again may be separated into an estimation and a quantization problem, as depicted in Fig. 2 (d). Observe that the quantizer is a maximum information rate rather than an MMSE quantizer. It contains the coder $\mathbf{F}^T\mathbf{R}_{xx}^{-1/2}$ and the decoder $\mathbf{R}_{xx}^{1/2}\mathbf{F}$. Thus, even though coder and decoder are inverses of each other, they are *non-orthogonal*, *unlike* the min-trace case. This establishes that for performance measures other than MSE orthogonal transformations can be outperformed by non-orthogonal transformations.

For jointly Gaussian message and measurements, the separation into Wiener filter and quantizer can also be deduced from Ephraim and Gray [10], using a result from Hua *et al.* [11]. Ephraim and Gray have generalized the result of [8], [9], detailed in (25), to more general performance measures, including weighted MSE. Hua *et al.* have shown that the min-det problem is equivalent to the weighted MMSE problem $\min_{\mathbf{A},\mathbf{B}} \operatorname{tr}(\mathbf{R}_{xx}^{-1}\mathbf{R}_{ee})$. Note that the discussion from Section III-A applies here, as well: Our proof shows that, using the AWN model and a transform coding system, it is optimum (under the maximum information rate criterion) to first obtain a *linear* MMSE estimate $\tilde{\mathbf{x}} = \mathbf{W}\mathbf{y}$ through the Wiener filter, and apply a maximum information rate quantizer to $\tilde{\mathbf{x}}$. It follows from [10] and [11] that, under the

maximum information rate criterion, it is generally optimum to obtain a *conditional mean* estimate of the message based on the observations, and then to quantize this estimate. For jointly Gaussian message and observations, the findings coincide.

In the proof that full canonical coordinates are optimum for maximum information rate quantization, the role of majorization is concealed through the use of Hadamard's and Minkowski's inequalities. For the min-trace problem $\min \mathrm{tr}\,(\mathbf{R}_{ee})$ of Section III-A, majorization requires maximum possible spread among the diagonal elements of $\mathbf{R}_{xy}\mathbf{R}_{yy}^{-1}\mathbf{R}_{xy}^{T}$. This leads to diagonalization of this matrix, with squared half canonical correlations on its diagonal. For the min-det problem of this section, we are minimizing $\mathrm{tr}\,(\mathbf{R}_{xx}^{-1/2}\mathbf{R}_{ee}\mathbf{R}_{xx}^{-T/2})$. Thus, majorization requires maximum possible spread among the diagonal elements of the squared coherence matrix $\mathbf{R}_{xx}^{-1/2}\mathbf{R}_{xy}\mathbf{R}_{yy}^{-1}\mathbf{R}_{xy}^{T}\mathbf{R}_{xx}^{-T/2}$. Again, achieving this maximum spread results in diagonalization of this matrix, with *squared full canonical correlations* on its diagonal.

V. Different Realizations

Rank reduction can be viewed as a special case of quantization since it amounts to assigning infinitely many bits to, say, r components and zero bits to the remaining $m-r$ components. However, optimality of canonical coordinates for rank reduction cannot be deduced from the results in the preceding sections, since assigning zero bits to components violates the high-resolution assumption. The only exception is the proof in Section III-B, which does not use the high-resolution assumption, but instead requires the coder to be of the form $\mathbf{A}=\mathbf{U}^{T}\mathbf{W}$, $\mathbf{U}$ orthogonal, and jointly Gaussian message and measurement. If we define the rate-distortion function as $f(0)=1$, $f(\infty)=0$, we have $f(b_i)=0$ for $i=1,...,r$, the components we keep, and $f(b_i)=1$ for $i=r+1,...,m$, the components we purge. Then the proof in Section III-B directly shows optimality of half canonical coordinates for rank reduction under the MMSE criterion, albeit under the restrictive assumptions mentioned above.

However, it has already been proved that some system of canonical coordinates is optimum for rank reduction in all generality, without making any restrictive assumptions at all. Again half canonical coordinates minimize the trace [12, p. 330] and full canonical coordinates minimize the determinant of the error covariance matrix [11]. Together with our results the important implications are these: Suppose we have a reduced rank Wiener filter, designed to either control MMSE or information rate. Then suppose this filter is to be quantized. The resulting reduced rank quantized structure retains the original coordinate system and replaces infinite precision internal coordinates with quantized coordinates. That is, *the coordinate system does not change*.

Fig. 2 displays several different implementations of reduced rank quantizers in full canonical coor-

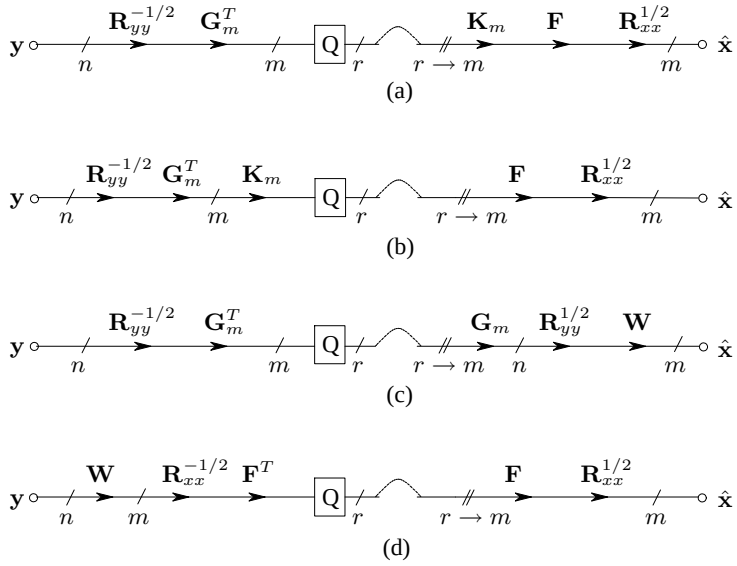


Fig. 2. Different implementations of reduced rank quantizers in full canonical coordinates

dinates. The quantizer Q can assign zero bits to a number of components, which means that only an r-dimensional statistic is stored or transmitted in these implementations. Before reconstruction, this r-dimensional statistic is augmented with zeros to build again an m-dimensional vector, as indicated in the figure by the $r \to m$ building block.

Each line of Fig. 2 is insightful. Lines (a) and (b) whiten with $\mathbf{R}_{yy}^{-1/2}$, resolve onto the basis for $< \mathbf{G}_m >$, quantize and filter (or vice versa), reconstruct in the basis $< \mathbf{F} >$, and color with $\mathbf{R}_{xx}^{1/2}$. There are implementations where explicit estimation precedes quantization and vice versa. For example, line (c) shows the quantized estimator to consist of whitening, analysis onto the basis for $< \mathbf{G}_m >$, quantizing, synthesis in the basis for $< \mathbf{G}_m >$, coloring, and filtering. In a storage or transmission application, only $\hat{\mathbf{u}} = \mathrm{Q}[\mathbf{G}_m^T \mathbf{R}_{yy}^{-1/2} \mathbf{y}]$ would be stored or transmitted, and $\mathbf{W}\mathbf{R}_{yy}^{1/2}\mathbf{G}_m$ would be computed at the receiver. Quantization in half canonical coordinates can be implemented very similarly, simply replacing $\mathbf{F}\mathbf{K}\mathbf{G}^T$ with $\mathbf{U}\mathbf{Z}\mathbf{V}^T$ and $\mathbf{R}_{xx}$ with the identity in Fig. 2.

VI. Conclusions

We have shown that transform coding of noisy sources is a story of majorization, either directly, or indirectly through the use of Hadamard, AM/GM, and Minkowski inequalities. We have proved that the right coordinate systems for quantization are the systems of half and full canonical coordinates. Half canonical coordinates minimize the trace and full canonical coordinates minimize the determinant of the error covariance matrix. It has been proved earlier [12, p. 330], [11], that canonical coordinates are

optimum for rank reduction, as well. Together with our results, this means that we can *first* choose a coordinate system and *then* decide how many bits to spend on how many components.

When looking at different transform coding schemes, it is essential to be very clear about the underlying assumptions. Our proofs that canonical coordinates are indeed optimum for transform coding require the use of the AWN model for quantization. Without at least the high resolution quantization assumption, very restrictive assumptions are needed to prove optimality of canonical coordinates. However, assumptions that are too restrictive usually limit performance: for instance, orthogonal transforms are in general inferior to non-orthogonal transforms for performance criteria other than MSE, in particular maximum information rate.

Finally, a remark regarding the extension to the complex case: It is often stated that quantization of complex vectors is essentially the same as for real vectors, as long as the definition of the inner product is changed from $< \mathbf{x}, \mathbf{y} >= \mathbf{x}^T\mathbf{y}$ to $< \mathbf{x}, \mathbf{y} >= \mathbf{x}^H\mathbf{y}$, where $\mathbf{x}^H$ is the complex conjugate transpose of $\mathbf{x}$. Thus, in this paper it should suffice to redefine covariance matrices as $\mathbf{R}_{xy} = E\mathbf{x}\mathbf{y}^H$ etc. This, however, is only true if message and observation are jointly proper, which means that the *complementary covariance matrices* $E\mathbf{x}\mathbf{x}^T$, $E\mathbf{x}\mathbf{y}^T$, and $E\mathbf{y}\mathbf{y}^T$ are all zero. If they are not, linear algebra will not give optimum performance and *widely linear* transformations must be used instead [20]. Canonical coordinates for improper complex random vectors are described in detail in [21].

ACKNOWLEDGEMENTS

We thank Stephen D. Voran and Tianjian Hu for their preliminary, unpublished, contributions to transform coders of noisy sources.

REFERENCES

[1] J. J. Y. Huang, P. M. Schultheiss, "Block Quantization of Correlated Gaussian Random Variables," *IEEE Trans. Commun. Syst.*, pp. 289 – 296, Sep. 1963

[2] V. K. Goyal, J. Zhuang, M. Vetterli, "Transform Coding with Backward Adaptive Updates," *IEEE Trans. Inform. Theory*, vol. 46, no. 4, pp. 1623 – 1633, July 2000

[3] A. Gersho, R. M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, Boston, MA, 1992

[4] P. P. Vaidyanathan, "Theory of Optimal Orthonormal Subband Coders," *IEEE Trans. Signal Processing*, vol. 46, no. 6, pp. 1528 – 1543, June 1998

[5] W. R. Bennett, "Spectra of Quantized Signals," *Bell Syst. Tech. J.*, vol. 27, pp. 446 – 472, July 1948

[6] B. Widrow, "A Study of Rough Amplitude Quantization by Means of Nyquist Sampling Theory," *IEEE Trans. Circuit Theory*, vol. 3, pp. 266 – 276, 1956

[7] K. Zeger, "Suboptimality of the Karhunen-Loève Transform for Fixed-Rate Transform Coding," *Proc. IEEE Globecom*, pp. 1224 – 1228, Nov. 2002

[8] J. K. Wolf, J. Ziv, "Transmission of Noisy Information to a Noisy Receiver With Minimum Distortion," *IEEE Trans. Inform. Theory*, vol. 16, no. 4, pp. 406 – 411, July 1970

[9] D. J. Sakrison, "Source Encoding in the Presence of Random Disturbance," *IEEE Trans. Inform. Theory*, pp. 165 – 167, Jan. 1968

[10] Y. Ephraim, R. M. Gray, "A Unified Approach for Encoding Clean and Noisy Sources by Means of Waveform and Autoregressive Model Vector Quantization," *IEEE Trans. Inform. Theory*, vol. 34, no. 4, pp. 826 – 834, July 1988

[11] Y. Hua, M. Nikpour, P. Stoica, "Optimal Reduced-Rank Estimation and Filtering," *IEEE Trans. Signal Processing*, vol. 49, no. 3, pp. 457 – 469, Mar. 2001

[12] L. L. Scharf, *Statistical Signal Processing*, Reading, MA: Addison-Wesley, 1991

[13] H. Hotelling, "Analysis of a Complex Pair of Statistical Variables into Principal Components," *J. Educ. Psychol.*, vol. 24, pp. 417 – 441, 498 – 520, 1933

[14] H. Hotelling, "Relations Between Two Sets of Variates," *Biometrika*, vol. 28, pp. 321 – 377, 1936

[15] M. L. Eaton, *Multivariate Statistics: A Vector Space Approach*, New York, NY: Wiely, 1983

[16] L. L. Scharf, J. K. Thomas, "Wiener Filters in Canonical Coordinates for Transform Coding, Filtering, and Quantizing," *IEEE Trans. Signal Processing*, vol. 36, no. 3, pp. 647 – 654, Mar. 1998

[17] L. L. Scharf, C. T. Mullis, "Canonical Coordinates and the Geometry of Inference, Rate, and Capacity," *IEEE Trans. Signal Processing*, vol. 48, no. 3, pp. 824 – 831, Mar. 2000

[18] A. W. Marshall, I. Olkin, *Inequalities: Theory of Majorization and Its Applications*, Academic Press, New York, NY, 1979

[19] R. A. Horn, C. R. Johnson, *Matrix Analysis*, Cambridge, UK: Cambridge University Press, 1985

[20] B. Picinbono, P. Chevalier, "Widely Linear Estimation with Complex Data," *IEEE Trans. Signal Processing*, vol. 43, no. 8, pp. 2030 – 2033, Aug. 1995

[21] P. J. Schreier, L. L. Scharf, "Second-Order Analysis of Improper Complex Random Vectors and Processes," *IEEE Trans. Signal Processing*, vol. 51, no. 3, pp. 714–725, Mar. 2003

CANONICAL COORDINATES ARE THE RIGHT COORDINATE SYSTEM FOR TRANSFORM CODING OF NOISY SOURCES

Peter J. Schreier[*], *Louis L. Scharf*[†], *Tianjian Hu*[‡], *Stephen D. Voran*[◦]

[*] Dept. of ECE, University of Colorado, Boulder, e-mail: peter@peter-schreier.com

[†] Dept's of ECE and Statistics, Colorado State University, Ft. Collins, e-mail: scharf@engr.colostate.edu

[‡] affiliation?

[◦] affiliation?

ABSTRACT

Historically, transform coding of noisy sources has been performed by first estimating the message and then quantizing this estimate. We show that it is also optimum to first transform the noisy observations into canonical coordinates, quantize, apply a Wiener filter in this coordinate system, and then transform the result back to the original coordinates. Canonical coordinates are uncorrelated, and quantization and Wiener filtering are applied to each component independently. Optimality of this approach can be proved assuming additive white quantization noise. Half canonical coordinates minimize the mean-squared error by minimizing the trace of the error covariance matrix and full canonical coordinates maximize information rate by minimizing the determinant of the error covariance matrix.

1. INTRODUCTION

In this paper we are interested in transform coding of noisy sources. Thus, we are looking for an answer to the question: Given a finite bit-budget of B bits, how can we most efficiently represent the information that a *noisy* observation $\mathbf{y} \in \mathbb{R}^n$ contains about a *random* message $\mathbf{x} \in \mathbb{R}^m$? A transform coder is depicted in Fig. 1. First, the observation $\mathbf{y}$ is passed through a linear transformation $\mathbf{A} \in \mathbb{R}^{m\times n}$, which we call the *coder*. The output of the coder is $\mathbf{u} = \mathbf{A}\mathbf{y}$, which is subsequently processed by a *scalar quantizer*. That is, each component of $\mathbf{u}$ is independently quantized. The quantizer output $\hat{\mathbf{u}}$ is supposed to be an efficient representation of the message $\mathbf{x}$, not the measurement $\mathbf{y}$. To produce an estimate $\hat{\mathbf{x}} = \mathbf{B}\hat{\mathbf{u}}$ of the message $\mathbf{x}$, $\hat{\mathbf{u}}$ is linearly transformed by the *decoder* $\mathbf{B} \in \mathbb{R}^{m\times m}$. Without loss of generality, we suppose that $m \leq n$. Furthermore, we will assume that $\mathbf{x}$ and $\mathbf{y}$ have zero mean and that we have the necessary second-order information available, namely, the covariance matrices of $\mathbf{x}$ and $\mathbf{y}$, denoted by $\mathbf{R}_{xx} = E\mathbf{x}\mathbf{x}^T$ and $\mathbf{R}_{yy} = E\mathbf{y}\mathbf{y}^T$, respectively, and the cross-covariance matrix $\mathbf{R}_{xy} = E\mathbf{x}\mathbf{y}^T$.

This work was supported by the DARPA ISP program under contract AFRL F33615-02-C-1198 and the 2001 NSF ITR Initiative under contract CCR0112573.

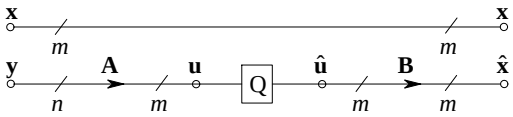


Fig. 1. Transform Coder

The problem can now be re-formulated as follows: First, how do we choose $\mathbf{A}$ and $\mathbf{B}$, i.e., in what coordinate system should we quantize? Second, how do we distribute the total number of bits B over the components of $\mathbf{u}$ so that $\hat{\mathbf{x}}$ is a good estimate of $\mathbf{x}$? To make precise what we mean by a "good" estimate we will employ two different performance measures: $E = \mathrm{tr}\,\mathbf{R}_{ee} = \mathrm{tr}\,E\mathbf{e}\mathbf{e}^T = E\|\hat{\mathbf{x}} - \mathbf{x}\|^2$, which is the mean squared error (MSE), and $V = \det \mathbf{R}_{ee}$, which measures the volume of the error covariance ellipsoid and thus information rate in the Gaussian case. For simplicity, let us refer to the problems where we try to minimize E and V as the *min-trace* and *min-det* problems, respectively.

In this paper, we show that a possible solution to the min-trace and the min-det problem is to first transform the noisy observations into a half canonical [3, p. 330] or full canonical coordinate system [4], respectively, quantize, Wiener filter in this coordinate system, and then transform the result back to the original coordinates. Canonical coordinates are uncorrelated, which means quantization and Wiener filtering are applied to each component independently. This extends previous results in that it provides a *concrete* coordinate system for quantization. Moreover, our results show that transform coders have many different implementations: for example, there are implementations where quantization precedes estimation, and vice versa.

The proofs of optimality that we provide are based on the additive white noise (AWN) model for quantization. If we let b_i denote the number of bits for quantizing component u_i, and $\sigma^2_{u_i} = Eu_i^2$ the variance of u_i, then the main assumptions of the AWN model may be summarized as follows:

$$\begin{aligned} E\mathbf{q}\mathbf{q}^T &= \mathrm{diag}\,(\sigma^2_{q_1}, \ldots, \sigma^2_{q_m}) \\ \sigma^2_{q_i} &= Eq_i^2 = c\sigma^2_{u_i} 2^{-2b_i}, \quad i = 1, \ldots, m \\ E\mathbf{u}\mathbf{q}^T &= \mathbf{0} \end{aligned}$$

The constant c is dependent on the distribution of u_i. If u_i is zero-mean Gaussian, then $c = \sqrt{3}\pi/2$.

2. MIN-TRACE PROBLEM

In this section, we show that for the min-trace problem the right coordinate system for quantization is the system of *half canonical coordinates* [3, p. 330]. Referring to the notation introduced in Fig. 1, we can express the error vector $\mathbf{e} = \hat{\mathbf{x}} - \mathbf{x}$ as

$$\mathbf{e} = (\mathbf{BAy} - \mathbf{x}) + \mathbf{Bq}.$$

If we invoke the AWN model, then $E\mathbf{xq}^T = \mathbf{0}$ and $E\mathbf{yq}^T = \mathbf{0}$, and the error covariance matrix $\mathbf{R}_{ee} = E\mathbf{ee}^T$ becomes

$$\mathbf{R}_{ee} = E\left[(\mathbf{BAy} - \mathbf{x})(\mathbf{BAy} - \mathbf{x})^T\right] + \mathbf{BR}_{qq}\mathbf{B}^T,$$

which can be expressed as the sum of three positive semi-definite terms:

$$\mathbf{R}_{ee} = \mathbf{Q} + (\mathbf{W} - \mathbf{BA})\mathbf{R}_{yy}(\mathbf{W} - \mathbf{BA})^T + \mathbf{BR}_{qq}\mathbf{B}^T \qquad (1)$$

In this equation, $\mathbf{W} = \mathbf{R}_{xy}\mathbf{R}_{yy}^{-1}$ is the Wiener filter and $\mathbf{Q} = \mathbf{R}_{xx} - \mathbf{R}_{xy}\mathbf{R}_{yy}^{-1}\mathbf{R}_{xy}^T$ is its filtering error covariance matrix. It is clear that we can make the middle term in (1) zero if we choose $\mathbf{BA} = \mathbf{W}$, and we will assume this optimum choice in what follows. Thus the infinite precision quantizer is a Wiener filter with error covariance $\mathbf{Q}$. Since $\mathbf{Q}$ does not depend on how we select $\mathbf{A}$ and $\mathbf{B}$, minimizing $\operatorname{tr}\mathbf{R}_{ee}$ amounts to minimizing $\operatorname{tr}\mathbf{BR}_{qq}\mathbf{B}^T$. This can be achieved with a variation on the proof for the noiseless case in [6, Appendix]. Denote the i-th column of $\mathbf{B}$ by $\mathbf{b}_i$. We then have

$$\begin{aligned}\operatorname{tr}\mathbf{BR}_{qq}\mathbf{B}^T &= \sum_{i=1}^{m} c2^{-2b_i}\sigma_{u_i}^2\|\mathbf{b}_i\|^2 \\ &\geq cm2^{-2b}\left[\prod_{i=1}^{m}\sigma_{u_i}^2\|\mathbf{b}_i\|^2\right]^{1/m}. \qquad (2)\end{aligned}$$

The inequality is an AM/GM inequality, with average bit rate $b = B/m$. Since $\sigma_{u_i}^2 = (\mathbf{AR}_{yy}\mathbf{A}^T)_{ii}$, Hadamard's inequality yields $\prod_{i=1}^{m}\sigma_{u_i}^2 \geq \det(\mathbf{AR}_{yy}\mathbf{A}^T)$. Using this inequality in (2) we obtain a new lower bound on $\operatorname{tr}\mathbf{BR}_{qq}\mathbf{B}^T$ as

$$cm2^{-2b}[\det(\mathbf{AR}_{yy}\mathbf{A}^T)\det(\mathbf{BB}^T)]^{1/m}\left[\frac{\prod_{i=1}^{m}\|\mathbf{b}_i\|^2}{(\det\mathbf{B})^2}\right]^{1/m}.$$

This expression can in turn be lower bounded by using another Hadamard inequality to arrive at

$$\begin{aligned}\operatorname{tr}\mathbf{BR}_{qq}\mathbf{B}^T &\geq cm2^{-2b}[\det(\mathbf{AR}_{yy}\mathbf{A}^T)\det(\mathbf{BB}^T)]^{1/m} \\ &= cm2^{-2b}[\det(\mathbf{R}_{xy}\mathbf{R}_{yy}^{-1}\mathbf{R}_{xy}^T)]^{1/m}.\end{aligned}$$

This final lower bound can be achieved if the inequalities we have used become equalities. For the Hadamard inequalities this means that

$$\mathbf{AR}_{yy}\mathbf{A}^T = \mathbf{D}_1 \qquad (3)$$

$$\mathbf{BB}^T = \mathbf{D}_2, \qquad (4)$$

where $\mathbf{D}_1$ and $\mathbf{D}_2$ are both $m \times m$ diagonal matrices. The two conditions (3) and (4) determine the *coordinate system* for $\mathbf{u}$. The AM/GM inequality, on the other hand, becomes an equality if

$$2^{-2b_i}\left(\mathbf{AR}_{yy}\mathbf{A}^T\right)_{ii}\|\mathbf{b}_i\|^2 = M, \qquad (5)$$

where the constant $M > 0$ is independent of i. This determines the *bit assignment* for $\mathbf{u}$.

Let us first talk about the coordinate system. From (3) and (4), and since $\mathbf{BA} = \mathbf{W}$, we find that $\mathbf{B}^T\mathbf{R}_{xy}\mathbf{R}_{yy}^{-1}\mathbf{R}_{xy}^T\mathbf{B} = \mathbf{D}_2^T\mathbf{D}_1\mathbf{D}_2^T$, which implies that $\mathbf{B}$ must diagonalize $\mathbf{R}_{xy}\mathbf{R}_{yy}^{-1}\mathbf{R}_{xy}^T$. We could thus choose $\mathbf{B}$ as the orthogonal matrix $\mathbf{U}$ from the eigenvalue decomposition $\mathbf{R}_{xy}\mathbf{R}_{yy}^{-1}\mathbf{R}_{xy}^T = \mathbf{U}(\mathbf{ZZ}^T)\mathbf{U}^T$, and $\mathbf{A} = \mathbf{U}^T\mathbf{R}_{xy}\mathbf{R}_{yy}^{-1} = \mathbf{U}^T\mathbf{W}$. With this choice, we are first estimating $\mathbf{x}$ by passing $\mathbf{y}$ through a Wiener filter $\mathbf{W}$, and then quantizing the estimate $\tilde{\mathbf{x}} = \mathbf{Wy}$, as in the noise-free case. The noisy quantization problem is thus reduced to a standard quantization problem, as long as we observe that the estimate $\tilde{\mathbf{x}}$ has covariance matrix $\mathbf{R}_{xy}\mathbf{R}_{yy}^{-1}\mathbf{R}_{xy}^T$ rather than $\mathbf{R}_{xx}$. This result connects to a finding in [2], where it was shown that for the MSE criterion it is optimum to apply the quantizer to a *conditional mean* estimator of the message based on the observations. The total MSE E is then the sum of the infinite precision filtering error and the error of quantizing the conditional mean estimate. Our result differs insofar as we have shown that, using the AWN model for quantization and a transform coding system, it is optimum to first obtain a *linear* MMSE estimate $\tilde{\mathbf{x}} = \mathbf{Wy}$ through the Wiener filter, and then quantize $\tilde{\mathbf{x}}$. Gaussianity is not required for our proof.

Moreover, our derivation also allows a different interpretation which brings fresh insight. If we start with the singular value decomposition (SVD) [3, p. 330]

$$\mathbf{R}_{xy}\mathbf{R}_{yy}^{-1/2} = \mathbf{UZV}^T = \mathbf{U}\begin{bmatrix}\mathbf{Z}_m & \mathbf{0}_{m\times(n-m)}\end{bmatrix}\begin{bmatrix}\mathbf{V}_m^T \\ \mathbf{V}_0^T\end{bmatrix},$$

one can check that taking $\mathbf{A} = \mathbf{ZV}^T\mathbf{R}_{yy}^{-1/2}$ and $\mathbf{B} = \mathbf{U}$ also satisfies (3) and (4). Thus $\mathbf{u} = \mathbf{ZV}^T\mathbf{R}_{yy}^{-1/2}\mathbf{y}$, with covariance $\mathbf{R}_{uu} = \mathbf{Z}_m\mathbf{Z}_m^T$, is quantized for $\hat{\mathbf{u}}$, and $\mathbf{x}$ is then estimated as $\hat{\mathbf{x}} = \mathbf{U}\hat{\mathbf{u}}$. The diagonal elements of $\mathbf{Z}_m$ are the *half canonical correlations* z_i between $\mathbf{x}$ and $\mathbf{y}$.

The MSE E will be minimized if bits are assigned according to (5), which says that

$$2^{-2b_i}z_i^2 = M, \qquad (6)$$

subject to $B = \sum_{i=1}^{m} b_i$. The solution to the bit assignment problem parallels the one for standard transform coding if

we observe that the variance of u_i is z_i^2:

$$b_i = b + \tfrac{1}{2}\log_2 \frac{z_i^2}{\left(\prod_{j=1}^m z_j^2\right)^{1/m}} \tag{7}$$

We can now express the MMSE in terms of z_i and M as

$$\min_{\mathbf{A},\mathbf{B}} E = \left[\operatorname{tr}\mathbf{R}_{xx} - \sum_{i=1}^m z_i^2\right] + cmM. \tag{8}$$

The first term in (8) accounts for the infinite-precision filtering error and the second term for the error due to quantization. If the bit rate B is given, then we use (7) to assign bits and (8) tells us the resulting MSE. On the other hand, if we want to achieve a given MSE, we can use (8) to determine the required M, and then (6) to assign bits.

Note that (3) and (4) allow the transformations $\mathbf{A}$ and $\mathbf{B}$ to be scaled by a non-singular diagonal matrix. If $\mathbf{A}$ is replaced by $\mathbf{D}^{-1}\mathbf{A}$ and $\mathbf{B}$ by $\mathbf{BD}$, then the quantizer input $\mathbf{u}$ will still be uncorrelated and according to (5) the optimum bit assignment is left unchanged. For instance, we could choose $\mathbf{A} = \mathbf{V}_m^T\mathbf{R}_{yy}^{-1/2}$ and $\mathbf{B} = \mathbf{U}\mathbf{Z}_m$. With this choice, the transformation $\mathbf{u} = \mathbf{A}\mathbf{y}$ takes $\mathbf{y}$ into a half canonical coordinate system, where the white, unit-variance, half canonical coordinates $\mathbf{u}$ are quantized. The transformation $\mathbf{B}$ applies a diagonal Wiener filter $\mathbf{Z}_m$ in canonical coordinates to the quantizer output $\hat{\mathbf{u}}$, and transforms the filter output back into the original coordinate system.

3. MIN-DET PROBLEM

In this section, we show that the right coordinate system for $\mathbf{u}$ to solve the min-det problem is the system of *full canonical coordinates* [4]. The proof will again be based on the AWN model. We start the minimization of $V = \det\mathbf{R}_{ee}$ by applying Minkowski's determinant inequality to (1):

$$V \geq \Big(\big(\det(\mathbf{Q} + \mathbf{B}\mathbf{R}_{qq}\mathbf{B}^T)\big)^{1/m} + \big(\det\big((\mathbf{W}-\mathbf{B}\mathbf{A})\mathbf{R}_{yy}(\mathbf{W}-\mathbf{B}\mathbf{A})^T\big)\big)^{1/m}\Big)^m$$

Since both the term in the left $\det(\cdot)$ and the right $\det(\cdot)$ expression are positive semi-definite, we can minimize V by making the second term zero, i.e., choosing $\mathbf{B}\mathbf{A} = \mathbf{W}$. We will assume this optimum choice in what follows. Using Minkowski's determinant inequality once more yields

$$V \geq \Big((\det\mathbf{Q})^{1/m} + \big(\det(\mathbf{B}\mathbf{R}_{qq}\mathbf{B}^T)\big)^{1/m}\Big)^m. \tag{9}$$

Since $\mathbf{Q}$ does not depend on the choice of $\mathbf{A}$ or $\mathbf{B}$, we must minimize $\det(\mathbf{B}\mathbf{R}_{qq}\mathbf{B}^T)$ in order to minimize the bound on V. Similar to the procedure in the previous section we have

$$\begin{aligned}\det(\mathbf{B}\mathbf{R}_{qq}\mathbf{B}^T) &= (c2^{-2b})^m \prod_{i=1}^m \left(\mathbf{A}\mathbf{R}_{yy}\mathbf{A}^T\right)_{ii} \det(\mathbf{B}^T\mathbf{B}) \\ &\geq (c2^{-2b})^m \det(\mathbf{A}\mathbf{R}_{yy}\mathbf{A}^T)\det(\mathbf{B}^T\mathbf{B}) \qquad (10)\\ &= (c2^{-2b})^m \det(\mathbf{R}_{xy}\mathbf{R}_{yy}^{-1}\mathbf{R}_{xy}^T).\end{aligned}$$

Inequality (10) is again Hadamard's inequality, which becomes an equality if $\mathbf{A}\mathbf{R}_{yy}\mathbf{A}^T$ is diagonal. Minkowski's inequality (9), on the other hand, becomes an equality if

$$\mathbf{Q} = K\cdot\mathbf{B}\mathbf{R}_{qq}\mathbf{B}^T$$

for some $K \geq 0$. With the knowledge that $\mathbf{A}\mathbf{R}_{yy}\mathbf{A}^T$ must be diagonal for equality, this means

$$\mathbf{Q} = cK\cdot\mathbf{B}[(\mathbf{A}\mathbf{R}_{yy}\mathbf{A}^T)\operatorname{diag}(2^{-2b_1},\ldots,2^{-2b_m})]\mathbf{B}^T. \tag{11}$$

The expression in square brackets is a diagonal matrix. Thus, in order to evaluate what (11) implies for $\mathbf{B}$, we would like to factor $\mathbf{Q} = \mathbf{T}\mathbf{D}_Q\mathbf{T}^T$, where $\mathbf{D}_Q$ is diagonal. To this end, we need the SVD of the coherence matrix [4]

$$\mathbf{R}_{xx}^{-1/2}\mathbf{R}_{xy}\mathbf{R}_{yy}^{-T/2} = \mathbf{F}\mathbf{K}\mathbf{G}^T = \mathbf{F}\begin{bmatrix}\mathbf{K}_m & \mathbf{0}_{m\times(n-m)}\end{bmatrix}\begin{bmatrix}\mathbf{G}_m^T \\ \mathbf{G}_0^T\end{bmatrix}.$$

Then we can re-write (11) as

$$\begin{aligned}\mathbf{Q} &= \mathbf{R}_{xx}^{1/2}\mathbf{F}(\mathbf{I}-\mathbf{K}\mathbf{K}^T)\mathbf{F}^T\mathbf{R}_{xx}^{T/2} \\ &= cK\cdot\mathbf{B}[(\mathbf{A}\mathbf{R}_{yy}\mathbf{A}^T)\operatorname{diag}(2^{-2b_1},\ldots,2^{-2b_m})]\mathbf{B}^T. \qquad (12)\end{aligned}$$

It is apparent that if we choose $\mathbf{A} = \mathbf{D}^{-1}\mathbf{K}\mathbf{G}^T\mathbf{R}_{yy}^{-1/2}$ and $\mathbf{B} = \mathbf{R}_{xx}^{1/2}\mathbf{F}\mathbf{D}$, where $\mathbf{D}$ is any non-singular diagonal matrix, then $\mathbf{B}\mathbf{A} = \mathbf{W}$, $\mathbf{A}\mathbf{R}_{yy}\mathbf{A}^T$ is diagonal, and (12) is satisfied. For $\mathbf{D} = \mathbf{K}_m$, the transformation $\mathbf{u} = \mathbf{A}\mathbf{y} = \mathbf{G}_m^T\mathbf{R}_{yy}^{-1/2}\mathbf{y}$ takes $\mathbf{y}$ into the full canonical coordinate system, where the white, unit-variance, full canonical coordinates $\mathbf{u}$ are quantized. The transformation $\mathbf{B} = \mathbf{R}_{xx}^{1/2}\mathbf{F}\mathbf{K}_m$ applies a diagonal Wiener filter $\mathbf{K}_m$ in full canonical coordinates to the quantizer output, and the filter output is transformed back into the original coordinate system with $\mathbf{R}_{xx}^{1/2}\mathbf{F}$. The diagonal Wiener filter $\mathbf{K}_m$ contains the *full canonical correlations* k_i between $\mathbf{x}$ and $\mathbf{y}$ on its diagonal.

Equation (12) also determines the right bit assignment strategy. Defining $\gamma_i^2 = k_i^2/(1-k_i^2)$, we must satisfy

$$2^{-2b_i}\gamma_i^2 = K, \quad i = 1,\ldots,m. \tag{13}$$

This means we have the same solution as for (6) with half canonical correlations z_i^2 replaced by γ_i^2,

$$b_i = b + \tfrac{1}{2}\log_2 \frac{\gamma_i^2}{\left(\prod_{j=1}^m \gamma_j^2\right)^{1/m}}. \tag{14}$$

We can now express the minimum achievable value of V in terms of k_i and K as

$$\min_{\mathbf{A},\mathbf{B}} V = \det(\mathbf{R}_{xx})\cdot\left(1+\frac{c}{K}\right)^m \cdot \prod_{i=1}^m (1-k_i^2). \tag{15}$$

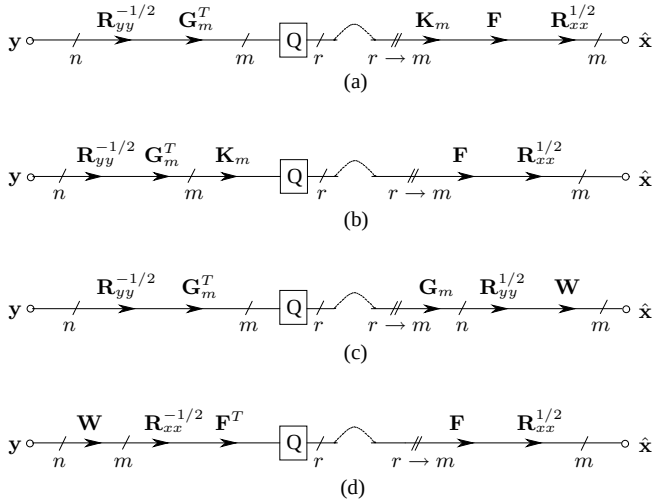


Fig. 2. Reduced rank quantizers, full canonical coordinates

Similar to the MMSE quantizer, if the bit rate B is given, then we use (14) to assign bits and (15) tells us the resulting V. On the other hand, if we want to achieve a given V, we can use (15) to determine the required K, and then (13) to assign bits.

Notice that, just like the min-trace problem, the min-det problem can also be solved by first computing the linear MMSE estimate of $\mathbf{x}$ as $\tilde{\mathbf{x}} = \mathbf{W}\mathbf{y}$ and then quantizing $\tilde{\mathbf{x}}$. To see this, write $\mathbf{A} = \mathbf{K}\mathbf{G}^T\mathbf{R}_{yy}^{-1/2} = \mathbf{F}^T\mathbf{R}_{xx}^{-1/2}\mathbf{R}_{xy}\mathbf{R}_{yy}^{-1} = \mathbf{F}^T\mathbf{R}_{xx}^{-1/2}\mathbf{W}$, and $\mathbf{B} = \mathbf{R}_{xx}^{1/2}\mathbf{F}$. Therefore, this problem again may be separated into an estimation and a quantization problem, as depicted in Fig. 2 (d). Observe that the quantizer is a maximum information rate rather than an MMSE quantizer. It contains the coder $\mathbf{F}^T\mathbf{R}_{xx}^{-1/2}$ and the decoder $\mathbf{R}_{xx}^{1/2}\mathbf{F}$. Thus, even though coder and decoder are inverses of each other, they are *non-orthogonal*, *unlike* the min-trace case.

4. DIFFERENT REALIZATIONS

It has already been proved that some system of canonical coordinates is optimum for rank reduction, even without the need to invoke the AWN model. Again half canonical coordinates minimize the trace [3, p. 330] and full canonical coordinates minimize the determinant of the error covariance matrix [1]. Together with our results the important implications are these: Suppose we have a reduced rank Wiener filter, designed to either control MMSE or information rate. Then suppose this filter is to be quantized. The resulting reduced rank quantized structure retains the original coordinate system and replaces infinite precision internal coordinates with quantized coordinates. That is, *the coordinate system does not change*.

Fig. 2 displays several different implementations of reduced rank quantizers in full canonical coordinates. The quantizer Q can assign zero bits to a number of components, which means that only an r-dimensional statistic is stored or transmitted in these implementations. Before reconstruction, this r-dimensional statistic is augmented with zeros to build again an m-dimensional vector, as indicated in the figure by the $r \to m$ building block.

Each line of Fig. 2 is insightful. There are implementations where explicit estimation precedes quantization and vice versa. For example, line (c) shows the reduced rank quantizer to consist of whitening, analysis onto the basis for $< \mathbf{G}_m >$, quantizing, synthesis in the basis for $< \mathbf{G}_m >$, coloring, and filtering. In a storage or transmission application, only $\hat{\mathbf{u}} = \mathrm{Q}[\mathbf{G}_m^T\mathbf{R}_{yy}^{-1/2}\mathbf{y}]$ would be stored or transmitted, and $\mathbf{W}\mathbf{R}_{yy}^{1/2}\mathbf{G}_m$ would be computed at the receiver. Quantization in half canonical coordinates can be implemented very similarly, simply replacing $\mathbf{F}\mathbf{K}\mathbf{G}^T$ with $\mathbf{U}\mathbf{Z}\mathbf{V}^T$ and $\mathbf{R}_{xx}$ with the identity in Fig. 2.

5. CONCLUDING REMARKS

Assuming the AWN model for quantization, we have proved that the right coordinate systems for quantization are the systems of half and full canonical coordinates. Half canonical coordinates minimize the trace and full canonical coordinates minimize the determinant of the error covariance matrix. It has been proved earlier [3, p. 330], [1], that canonical coordinates are optimum for rank reduction, as well. Together with our results, this means that we can *first* choose a coordinate system and *then* decide how many bits to spend on how many components. More details can be found in the journal version [5] of this paper.

6. REFERENCES

[1] Y. Hua, M. Nikpour, P. Stoica, "Optimal Reduced-Rank Estimation and Filtering," *IEEE Trans. Signal Processing*, vol. 49, no. 3, pp. 457 – 469, Mar. 2001

[2] D. J. Sakrison, "Source Encoding in the Presence of Random Disturbance," *IEEE Trans. Inform. Theory*, pp. 165 – 167, Jan. 1968

[3] L. L. Scharf, *Statistical Signal Processing*, Reading, MA: Addison-Wesley, 1991

[4] L. L. Scharf, J. K. Thomas, "Wiener Filters in Canonical Coordinates for Transform Coding, Filtering, and Quantizing," *IEEE Trans. Signal Processing*, vol. 36, no. 3, pp. 647 – 654, Mar. 1998

[5] P. J. Schreier, L. L. Scharf, "Canonical Coordinates for Transform Coding of Random Sources from Noisy Observations," submitted to *IEEE Trans. Signal Processing*

[6] P. P. Vaidyanathan, "Theory of Optimal Orthonormal Subband Coders," *IEEE Trans. Signal Processing*, vol. 46, no. 6, pp. 1528 – 1543, June 1998

Sensor Scheduling for Target Tracking: A Monte Carlo Sampling Approach ⋆

Ying He

Institute for Systems Research, 2231 AV Williams Building, University of Maryland, College Park, MD 20742, USA

Edwin K. P. Chong *

Department of Electrical and Computer Engineering, Colorado State University, 1373 Campus Delivery, Fort Collins, CO 80523-1373, USA

Abstract

We study the problem of sensor-scheduling for target tracking—to determine which sensors to activate over time to trade off tracking performance with sensor usage costs. We approach this problem by formulating it as a partially observable Markov decision process (POMDP), and develop a Monte Carlo solution method using a combination of particle filtering for belief-state estimation and sampling-based Q-value approximation for lookahead. To evaluate the effectiveness of our approach, we consider a simple sensor-scheduling problem involving multiple sensors for tracking a single target.

1 Introduction

One of the key problems in the design and operation of modern tracking systems is sensor scheduling, which aims to improve tracking system performance, utilize limited system resources more effectively and efficiently, and offer much faster adaptation to changing environments [1]. The basic problem is to select which sensors to activate for target tracking over time to trade off tracking performance with sensor usage costs.

⋆ This work was partially supported by DARPA under contracts F33615-02-C1198 and FA9550-04-1-0371, and by NSF under grants ECS-0098089, ANI-0099137, and ANI-0207892. The views and conclusions in this document are those of the authors and should not be interpreted as representing the official policies of DARPA, NSF, or the U.S. Government.

* Edwin K. P. Chong is the corresponding author.

Email addresses: yhe@umd.edu (Ying He), echong@engr.colostate.edu (Edwin K. P. Chong).

A number of papers have addressed the sensor-scheduling problem for different applications. In [2–4], this problem is formulated as an optimization problem to minimize instantaneous estimation errors and/or maximize information gains. In such schemes, however, long-term performance is not considered, which leads to myopic sensor-scheduling policies.

To incorporate long-term performance measures, the sensor-scheduling problem may be framed as a stochastic optimal control problem. In this case, a partially observable Markov decision process (POMDP) framework is a natural approach, which is able to address both short-term and long-term benefits and costs [5–8].

Within the POMDP framework, the process measured by sensors (target position, velocity, etc.) is a Markov process, and sensor scheduling is based on recursively estimating and updating the *belief state*, the posterior distribution of the process given the history of the sensor measurements and the sensor-scheduling actions. In some situations, the process dynamics and measurements can be represented as linear Gaussian state-space models, in which case the belief state can be calculated analytically by Kalman filtering. In other situations [7,8], the process dynamics can be modeled as a partially observable, finite state Markov chain, and it is also feasible to obtain an analytic solution for the belief state using a hidden Markov model (HMM) filter. In practice, however, process dynamics and observations can be very complicated—usually nonlinear, non-Gaussian, and high-dimensional—which precludes analytic solutions.

In this paper, we propose to explore the sensor-scheduling problem within a POMDP framework but without relying on analytic expressions for belief states. Instead, we develop a Monte Carlo solution approach that combines particle filtering for non-Gaussian, nonlinear belief-state estimation, and a Q-value approximation method for solving the POMDP via "lookahead." Our goal is to design a policy for sensor scheduling to manage (simultaneously) tracking performance and sensor usage. The Q-value approximation method aims to deal with the issue that the state space for the POMDP model can be very large, practically ruling out the use of methods that rely on direct reasoning with the state space in computing an optimal policy.

Particle filtering is a promising Monte Carlo method for posterior-distribution estimation, working with random samples drawn from the process distribution. The Q-value approximation method involves computing, for each candidate action to be taken, a value of the "cost" of that action (the Q-value) and selecting the action with minimum cost. Our approach blends the two separate techniques in a natural way. The particle filter provides the Q-value approximation method a set of states (particles) to initiate the evaluation processes, and in return the Q-value approximation method delivers updated actions for the particle filtering belief-state estimation.

Our approach benefits from several appealing features. First, it can take both long-term and short-term costs and benefits into account. Second, because it is a Monte Carlo method, which does not rely on analytical tractability, it is straightforward in our approach to incorporate sophisticated models for sensor behavior and target dynamics. In particular, the model we introduce in Section 3 includes a non-

linear observation map and sensors with blind zones.

Our main contribution here is to combine POMDPs with particle filtering for sensor scheduling. The formulation of the sensor-scheduling problem as a POMDP is itself not new (see, e.g., [7,8]). However, our use of Monte Carlo sampling methods for Q-value approximation is new in the sensor-scheduling context. Moreover, in previous work, the belief state was estimated and updated using either Kalman filtering or HMM filtering (e.g. [7,8]). Both Kalman filtering and HMM filtering are inadequate for nonlinear, non-Gaussian state estimation. Recently, the authors of [4,9,10] have also studied the use of particle filtering for sensor-scheduling. However, in [4,9,10] the problem was not formulated as a POMDP.

To evaluate the effectiveness of our approach, we study a simple sensor-scheduling problem involving multiple sensors for tracking a single target. In particular, we explore the tradeoff between tracking performance and sensor usage costs. Our simulation results demonstrate that our method of combining particle filtering with Q-value approximation is effective in calculating a sensor-scheduling policy that systematically allows trading off tracking performance for sensor usage costs.

2 Preliminaries

We begin with a brief description of POMDPs; we follow the treatment in [11]. A POMDP is specified by state space S, action space U, observation space Z, state transition law $K(s'|s,u)$ ($s \in S$ and $u \in U$), observation map $L(z|u,s)$ ($z \in Z$), initial state distribution p_0, and one-step cost function $g(s,u)$. The POMDP generates a sequence of states that evolves as follows. At time $k = 0$, the system starts at the initial (unobservable) state s_0 with the given initial distribution p_0. If at time k, the state of the system is s_k and control u_k (chosen from a set of available actions $U(s_k)$) is applied, a cost $g(s_k, u_k)$ is incurred and the system moves to state s_{k+1} according to the transition law $K(s_{k+1}|s_k, u_k)$; observation z_{k+1} is generated according to the observation map $L(z_{k+1}|u_k, s_{k+1})$.

A *policy* for the POMDP can be defined as a sequence $\pi = \{\mu_k(p(s_k|I_k))\}$ such that, for each k, $\mu_k(p(s_k|I_k))$ is a state-feedback map that specifies an action u_k on U depending on the *belief state* $p(s_k|I_k)$, the posterior probability distribution of state s_k conditioned on the *observable history* I_k ($I_0 := (p_0)$ and $I_k := (p_0, u_0, z_1, \ldots, z_{k-1}, u_{k-1}, z_k)$ for $k \geq 1$). We can track belief states in a POMDP using Bayes' rule.

Let $J_H(p(s_0|I_0)) = E\left(\sum_{k=0}^{H-1} \int g(s_k,u_k)\, dp(s_k|I_k)\right) = E\left(\sum_{k=0}^{H-1} g(s_k,u_k)\right)$ be the expected total cost over a horizon of H time steps, where the expectation is taken over all possible belief-state sequences. Our objective is to find a policy $\pi^* = \{\mu_0^*(p(s_0|I_0)), \mu_1^*(p(s_1|I_1)), \ldots\}$ that minimizes $J_H(p(s_0|I_0))$. We denote the associated *optimal value* (a function of the initial belief state) by $J_H^*(p(s_0|I_0))$. We write $Q_H(p(s|I),u) = \int g(s,u)\, d(p(s|I)) + E[J_{H-1}^*(p(s'|I'))]$ (called the *Q-value*), where $J_{H-1}^*(p(s'|I'))$ is the optimal value over $H-1$ time steps starting at the "next" belief state $p(s'|I')$. It turns out that an optimal policy satisfies $\mu_k^*(p(s_k|I_k)) =$

$\arg\min_{u_k \in U(s_k)} Q_{H-k}(p(s_k|I_k), u_k)$, where $J^*_H(p(s_0|I_0)) = \min_{u \in U(s_0)} Q_H(p(s_0|I_0), u)$ (also called Bellman's optimality equation for POMDP); see [11] for further details.

We assume H to be very large, so that the optimal policy can be assumed to be stationary. In this case, the optimal policy is approximated by assuming, at each time, that the remaining horizon is H steps, so that the optimal action at time k can be taken to be $u^*_k = \arg\min_{u \in U(s_k)} Q_H(p(s_k|I_k), u)$. This approach is called *receding horizon control*, common in the optimal control literature (e.g., [12]). Note that the resulting control law is simply this: given the current state, choose the action that minimizes the Q-value at that state. Because the Q-value of an action summarizes the future impact of taking that action, our control approach is also called "lookahead."

3 Sensor-Scheduling for Target-Tracking Problem Formulation

In our context, there are many sensors distributed in a sensor field to track targets, and a global processor processing data from all sensors. For homogeneous sensors, the tracking accuracy can be improved through data fusion of multiple sensors. However, the larger the number of sensors, the more resources they consume. Therefore it is necessary to select an appropriate number of sensors to balance between tracking accuracy and sensor usage. For heterogeneous sensors, their sensor-usage characteristics and/or the quality of the data they transmit to the global processor are different. In this case, data from one sensor can be used to complement the data from other sensors in order to obtain broader coverage and more accurate target-state estimates. How to appropriately select the right sensor combination to reach a tradeoff between tracking accuracy and sensor usage is a key task of sensor scheduling.

We now describe a formulation of the sensor-scheduling problem within a POMDP framework. Although our approach is fairly general, for ease of presentation we make some simplifying assumptions:

- We only track a single target;
- The target states to be tracked consist of its two-dimensional position and velocity;
- There are M sensors located at fixed positions to measure the following parameters: range, range rate, and azimuth of the target;
- At each time step, only one sensor is selected (activated).

We consider an aggregate tracking system state vector $s_k = [t_k, a_k]^T$, where t_k and a_k are summaries for the target features and the sensor operations, respectively, in the tracking system sufficient to characterize the objectives and potential actions. Specifically, $s_k = [\underbrace{x_k, \dot{x}_k, y_k, \dot{y}_k}_{t_k}, \underbrace{a_{k,1}, \ldots, a_{k,M}}_{a_k}]^T$, where x_k and y_k are the target-position Cartesian coordinates, $\dot{x}_k$ and $\dot{y}_k$ are velocities, and $a_{k,m} \in \{0, 1\}$ is the activity status of sensor m, $m = 1, \ldots, M$. The action space U is $\{1, \ldots, M\}$, and action $u_k \in \{1, \ldots, M\}$ represents the sensor selected at time k. The observation at time k is $z_k = [d_k, r_k, \theta_k, \dot{r}_k]^T$, where $d_k \in \{0, 1\}$ represents successful detection, and r_k, θ_k, and $\dot{r}_k$ are range, azimuth, and range-rate measurements of the target using the selected sensor u_k at time k (if $d_k = 0$, then r_k, θ_k, and $\dot{r}_k$ are ignored).

In our formulation, the transition law $K(s'|s,u)$ and observation map $L(z|u,s)$ are defined by the state equation and the observation equation as follows:

$$s_{k+1} = f(s_k, u_k, \nu_k) \tag{1}$$
$$z_k = h(s_k, \omega_k), \tag{2}$$

where f and h represent the state dynamics and the observation map, respectively, and ν_k and ω_k represent the randomness in state transitions and observations, respectively. We assume that $\{\nu_k\}$ and $\{\omega_k\}$ are mutually independent i.i.d. random variables with distributions p_ν and p_ω. Then, $K(ds'|s,u) = \int \mathbf{1}_{ds'}(\nu') f(s,u,\nu')\, p_\nu(d\nu')$ and $L(dz|u,s) = \int \mathbf{1}_{dz}(\omega') h(s,\omega')\, p_\omega(d\omega')$ ($\mathbf{1}_A$ represents the indicator function of A).

We first describe the state dynamics f. Because the state vector s_k is composed of two segments, the state dynamics can be decomposed in the following way: $f(s_k, u_k, \nu_k) = [f^t(t_k, \nu_k^t), f^a(u_k)]^T$, where ν_k^t represents target motion uncertainties. The form of $f^a(u_k)$ is clear: all its components are 0 except for the component corresponding to the selected sensor u_k, where it is 1. The specific form of f^t represents the model for the motion of the target. As an example, the particular model used in our simulation experiments is as follows (taken from [13]):

$$t_{k+1} = \begin{bmatrix} x_{k+1} \\ \dot{x}_{k+1} \\ y_{k+1} \\ \dot{y}_{k+1} \end{bmatrix} = f^t(t_k, \nu_k^t) = \begin{bmatrix} 1 & T_s & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ \dot{x}_k \\ y_k \\ \dot{y}_k \end{bmatrix} + \begin{bmatrix} \frac{T_s^2}{2} & 0 \\ T_s & 0 \\ 0 & \frac{T_s^2}{2} \\ 0 & T_s \end{bmatrix} \begin{bmatrix} \nu_k^x \\ \nu_k^y \end{bmatrix} \tag{3}$$

where T_s is the sampling interval (assumed constant), and ν_k^x and ν_k^y are independent noise processes with zero mean and variances σ_x^2 and σ_y^2.

Next, we describe the observation map h, which represents how the sensor measurements depend on the state. The particular form of h depends on the type of sensors being considered. For example, in our simulation experiments, we follow the radar model of [13], where each sensor has a Doppler blind zone (as is the case with a CW or pulse Doppler radar). The probability of detection according to this model is (for a particular sensor m):

$$\begin{cases} P_d(m), \text{ if } \left| \frac{(x_k - sp_x(m))\cdot \dot{x}_k + (y_k - sp_y(m))\cdot \dot{y}_k}{\sqrt{(x_k - sp_x(m))^2 + (y_k - sp_y(m))^2}} \right| \geq B_0(m), \\ 0, \qquad \text{otherwise}, \end{cases} \tag{4}$$

where $P_d(m) \in (0,1]$, $B_0(m)$ is the limit of the Doppler blind zone for sensor m, and $sp_x(m)$ and $sp_y(m)$ are the Cartesian coordinates of the fixed position of sensor m.

For this example, the observation map is given by

$$h(s_k, \omega_k) = \begin{bmatrix} h_k^d(s_k, \omega_k^d) \\ \sqrt{(x_k - sp_x(m(a_k)))^2 + (y_k - sp_y(m(a_k)))^2} + \omega_k^r(m(a_k)) \\ \tan^{-1} \frac{y_k - sp_y(m(a_k))}{x_k - sp_x(m(a_k))} + \omega_k^\theta(m(a_k)) \\ \frac{(x_k - sp_x(m(a_k))) \cdot \dot{x}_k + (y_k - sp_y(m(a_k))) \cdot \dot{y}_k}{\sqrt{(x_k - sp_x(m(a_k)))^2 + (y_k - sp_y(m(a_k)))^2}} + \omega_k^{\dot{r}}(m(a_k)) \end{bmatrix} \tag{5}$$

where $h_k^d(s_k, \omega_k^d)$ is given by,

$$\begin{cases} \mathbf{1}_{\{\omega_k^d > P_d(m(a_k))\}}, \text{ if } \left| \frac{(x_k - sp_x(m(a_k))) \cdot \dot{x}_k + (y_k - sp_y(m(a_k))) \cdot \dot{y}_k}{\sqrt{(x_k - sp_x(m(a_k)))^2 + (y_k - sp_y(m(a_k)))^2}} \right| \geq B_0(m(a_k)), \\ 0, \qquad \text{otherwise,} \end{cases} \tag{6}$$

$m(a_k)$ is the currently selected sensor, ω_k^d is uniformly distributed over $(0, 1)$, and $\omega_k^r(m)$, $\omega_k^\theta(m)$, and $\omega_k^{\dot{r}}(m)$ represent independent observation noise processes with zero mean and variances $\sigma_r^2(m)$, $\sigma_\theta^2(m)$, and $\sigma_{\dot{r}}^2(m)$, respectively, $m = 1, \ldots, M$.

The one-step cost function $g(s_k, u_k)$ is an integrated metric that accounts for the target-tracking performance and the sensor-usage costs at time-step k. As an example, in our simulation experiments we use

$$g(s_k, u_k) = E\left[\|\hat{t}_k - t_k\|^2\right] + \sum_{m=1}^{M} \left(c_m^{usage} \cdot \mathbf{1}_{\{u_k = m\}} + c_m^{start} \cdot (a_{k,m} - \mathbf{1}_{\{u_k = m\}})^2 \right), \tag{7}$$

where $\hat{t}_k$ is the estimated value of state segment t_k (we use the mean of $p(t_k|I_k)$ as $\hat{t}_k$), and c_m^{usage} and c_m^{start} are the unit power-consumption cost and the unit sensor start-up power-consumption cost for sensor m, respectively. The parameters c_m^{usage} and c_m^{start}, which control the tradeoff between tracking error and sensor usage costs, are assumed to be user-specified. Certainly one can imagine setting up a system to tune such parameters on-line, for example in response to measurements. However, the criterion that drives this kind of tuning must then be specified; again, we consider such criteria to be user-specified. But then this scenario falls right back into our original framework, except with a more complicated objective function.

So far we have defined the state vector s_k, the action u_k, the observation vector z_k, the state transition law $K(s'|s, u)$, the observation map $L(z|u, s)$, and the one-step cost $g(s_k, u_k)$ for the sensor-scheduling POMDP model. Next we show how to obtain an approximate optimal policy to schedule sensors for target tracking.

4 POMDP Solution via a Combination of Particle Filtering and Q-value Approximation

In this section, we present our control approach based on "lookahead" for solving the sensor-scheduling POMDP, which leads to an approximate optimal policy. The

policy specifies, for each possible belief state, the (approximate) best sensor-activating action to implement according to the objective function, where the belief state here is the posterior distribution of the tracking system state conditioned on the observable history at each time.

Recall from Section 2 that in the "lookahead" approach, the action is chosen at each decision time by minimizing the Q-value for a moving horizon into the future. To be precise, $u_k^* = \arg\min_{u \in U(s_k)} = Q_H(p(s_k|I_k), u)$, where $p(s_k|I_k)$ is the current belief state.

Note that the action u_k^* to be chosen depends on belief state $p(s_k|I_k)$. Under certain circumstances, analytical expressions of belief states can be derived. For instance, if the observable history I_k is linear-Gaussian with respect to the tracking system state s_k, we can derive an analytical expression to recursively estimate the posterior distribution with a Kalman filter. Alternatively, if the tracking system state can be modeled as a Markov chain, as in [7,8], an HMM filter can be used for analytical belief-state estimation. In practice, however, the relationship between the tracking system state and the observable history can be very complex—usually nonlinear, non-Gaussian, and high dimensional—which makes it impossible to obtain an analytical solution.

To overcome this difficulty, we describe a novel general approach that combines two techniques: particle filtering for belief-state estimation and Q-value approximation, in which we represent the belief state by a cloud of particles. The Q-value approximation method addresses the issue that the state space in practice can be very large (especially in light of the need to represent a belief state), precluding the use of methods that rely on direct reasoning with the state space in computing an optimal policy.

4.1 Particle Filtering for Belief-State Estimation

Particle filtering is a sequential Monte Carlo method for on-line learning within a Bayesian framework [14]. The method works with random samples drawn from the underlying distribution, and is computationally realizable even for high-dimensional problems. Particle filtering allows the use of realistic models, incorporation of a priori information, and integration with decision processes.

In most particle-filtering formulations [14], the state equation, observation equation, and the initial state probability are described by $s_{k+1} = f(s_k, v_k)$, $z_k = h(s_k, \omega_k)$, and $p(s_0) = p_0$, respectively, and the goal is to estimate recursively the posterior distribution $p(s_k|z_1, z_2, \ldots, z_k)$. However, in our sensor-scheduling problem, we have a control variable u_k in the state equation (1), and our goal is to estimate the posterior distribution $p(s_k|I_k) = p(s_k|p_0, u_0, z_1, \ldots, z_{k-1}, u_{k-1}, z_k)$. Particle filtering with control variables has been discussed recently in [15], though not within a POMDP framework.

We can write the approximation of the posterior distribution $p(s_k|I_k)$ by a set of samples or particles: $\hat{p}_N(ds_k|I_k) = \sum_{i=1}^{N} w_k^{(i)} \delta_{s_{pt,k}}(ds_k)$, where $\delta_{s_{pt,k}}$ denotes the Dirac-

delta mass located at $s_{pt,k}$, N is the number of particles, and $w_k^{(i)}$ are the normalized "importance" weights.

Following [14], our particle filtering algorithm to estimate $p(s_k|I_k)$ is as follows:

(1) *Initialization*, $k = 0$.
- For $i = 1, \ldots, N$, sample $s_0^{(i)} \sim p(s_0)$, set $\tilde{w}_0^{(i)} = 1/N$, and set $k = 1$.

(2) *Importance-sampling step.*
- For $i = 1, \ldots, N$, sample $\tilde{s}_k^{(i)} \sim q(s_k|s_{k-1}^{(i)}, I_k)$, where $q(s_k|s_{k-1}, I_k)$ is a preselected "proposal" function.
- For $i = 1, \ldots, N$, update the importance weights

$$\tilde{w}_k^{(i)} = \tilde{w}_{k-1}^{(i)} \frac{K(\tilde{s}_k^{(i)}|\tilde{s}_{k-1}^{(i)}, u_{k-1}) L(z_k|u_{k-1}, \tilde{s}_k^{(i)})}{q(\tilde{s}_k^{(i)}|\tilde{s}_{k-1}^{(i)}, I_k)}.$$

- Normalize the importance weights according to $w_k^{(i)} = \frac{\tilde{w}_k^{(i)}}{\sum_{j=1}^{N} \tilde{w}_k^{(j)}}$.

(3) *Selection Step.*
- Resample with replacement N particles $(s_k^{(i)}; i = 1, \ldots, N)$ from the set $(\tilde{s}_k^{(i)}; i = 1, \ldots, N)$ according to the normalized importance weights.
- Set $k \leftarrow k + 1$ and go to step 2.

Note that the main difference between our algorithm and the standard algorithm is that the importance-sampling step in our algorithm involves the probability distribution of observation conditioned on the action.

In the importance-sampling step, we often choose either $p(s_k|s_{0:k-1}, I_k)$ or $p(s_k|s_{k-1}, u_{k-1})$ as the "proposal" function. In the selection step, many schemes have been proposed, such as residual sampling, systematic sampling, and Markov chain Monte Carlo (MCMC) [14].

For the example in our simulation experiments, we use a special particle filter algorithm for belief state estimation that exploits a priori knowledge of the sensor blind zones. The basic idea is to have one set of particles for sensor blind zones, and the other set of particles for the target. Though this idea is similar to that in [13], our particle filter is designed for belief-state estimation rather than ordinary posterior distribution estimation. We omit the details for brevity.

4.2 Q-value Approximation with Particle Filtering

Recall that according to the "lookahead" procedure, the action at time k is chosen as $u_k^* = \arg\min_{u \in U(s_k)} = Q_H(p(s_k|I_k), u)$. We now describe how we approximate the Q-values for the candidate actions. The need to approximate the Q-values stems from the intractability of computing precise Q-values due to the excessively huge state space in practice.

Several Q-value approximation methods have been proposed for large state-space MDPs [16,17]. Here we consider one particular Q-value approximation method—*policy*

rollout [16]. In the policy-rollout method, we estimate the Q-value for each belief state and each action by averaging the evaluated accumulated costs from several Monte Carlo simulation runs using a given *base policy*. This approximation gives us an upper bound on the true Q-value.

Because we represent the belief state $p(s_k|I_k)$ using a cloud of particles, our Q-value approximation method can take advantage of this representation in initiating Monte Carlo simulation runs of the base policy. Specifically, we start each simulation run at one particle for the belief state, apply action u for the first time step, and apply a given base policy π_b for the remaining time steps. This allows us to generate N simulation runs, one for each of the N different particles. We estimate the Q-value for each belief state and each action by averaging the evaluated accumulated costs from these N Monte Carlo simulation runs. The resulting rollout policy is the action minimizing $\hat{Q}_H(\hat{p}_N(s_k|I_k), u) = \frac{1}{N}\sum_{i=1}^{N}\{g(s_k^{(i)}, u) + \hat{J}_{H-1}^{\pi_b}(Y^{(i)})\}$, where $Y^{(i)}$ is the state after the first time step (for particle i). Usually, we choose as the base policy a heuristic policy that is known to be reasonable. The choice of a base policy may have a significant impact on the performance of the rollout policy. For more details on how properly to choose a base policy, see [16].

4.3 On the Computational Burden of Our Approach

A primary concern in applying sophisticated methods, such as ours, is the computational burden involved. It is instructive to compare the computational burden of our scheme with that of conventional myopic schemes, such as CPA (Closest Point of Approach), which we will use in our simulation experiments for comparison. As the basis of this comparison, we first note that the computational requirements in our approach stem from three sources:

(1) the particle-filter algorithm for belief-state estimation,
(2) the selection of an action with minimum Q-value, and
(3) simulation runs for Q-value calculations.

The computational burden involved in item 1 above is required of any tracking method that uses particle filters, including myopic approaches such as CPA. The extent of this burden depends on the number of particles used in the filter. Of course, with additional assumptions, the particle-filtering approach can be replaced by some other, such as Kalman filtering (but this consideration applies to both our approach and conventional approaches).

Item 2 above, which involves solving an optimization problem, is common to both our approach and myopic approaches. The difference between the two is that the objective function being optimized in our approach is given by the Q-values, whereas in myopic approaches the objective function is some given (myopic) criterion. In the case of CPA, this function is given by the distance between sensors and the estimated target position. The fact that our approach involves solving an optimization problem at every step, just like in myopic approaches, is a desirable consequence of Bellman's optimality equation (see Section 2). In either case, the computational burden required

for this optimization depends on the size of the search space (the number of feasible actions).

The third source of computational requirements in our approach is that of evaluating Q-values via Monte Carlo sampling. This is a computational burden that our method has to bear, but one that is not present in conventional myopic methods involving "simple" objective functions (such as CPA). A distinct advantage of Monte Carlo sampling is that we can incorporate sophisticated objective functions, taking into account factors that are not possible to account for analytically. Hence, myopic methods with complicated objective functions that are impossible to evaluate analytically may also take advantage of Monte Carlo methods. In this case, the computational burden becomes comparable to that of our method. Similarly, in (rare) situations where the Q-values in our method can be computed analytically, the burden of Monte Carlo sampling may be ameliorated. In either case, the computational requirement in Monte Carlo sampling depends on the length of the simulation runs and the number of samples needed for averaging. By controlling these quantities, we can trade off the performance of the resulting policy for reduced computational complexity.

5 Simulation Experiments

We evaluated our approach via simulation experiments. In our experiments, the target-motion model is given by equation (3), with $\sigma_x = \sigma_y = g\ T_s^{-1/2}$ (g is the acceleration of gravity), and there are $M = 4$ sensors available (sensor A, B, C, and D with $m = 1$, 2, 3, and 4), with the observation map (5). The other parameters used in our experiments are as follows: the sampling interval is $T_s = 2$ sec; the locations of sensors are $\{(sp_x(1), sp_y(1)), (sp_x(2), sp_y(2)), (sp_x(3), sp_y(3)), (sp_x(4), sp_y(4))\} = \{(0, 0), (-10, 30), (0, 60), (10, 30)\}$ (km, km); the limits of the Doppler blind-zone for all sensors are $B_0 = 100$ km/h; and the probabilities of detection for the sensors are all equal: $P_d = 0.9$.

We compare the performance of our rollout policy to the commonly used CPA (Closest Point of Approach) policy. CPA, selecting the closest sensor to the target estimate, is a "greedy" approach that does not take into account the sensor power consumption or the sensor error statistics. We consider two scenarios. In scenario **1**, we assume that one of the sensors consumes much more energy than the other three, and that the error statistics for all the sensor measurements are the same: $\sigma_r = 250$ m, $\sigma_{\dot{r}} = 3$ m/s, and $\sigma_\theta = 1^o$. Because our rollout policy takes this information into account, it can avoid selecting the more costly sensor at appropriate times. Figures 1 and 3(a) illustrate the true trajectories of the target, the estimates of the target positions, and the sequences of the selected sensors using the CPA policy and the rollout policy, respectively. Figure 2(a) shows the accumulated tracking error and power-consumption cost from the CPA and rollout policies. Here, sensor C is the sensor that consumes much more energy than the other three.

In scenario **2**, we assume that all sensors have the same power consumption but

sensor B has much smaller measurement noise than others with its error statistics being: $\sigma_r = 50$ m, $\sigma_{\dot{r}} = 0.6$m/s, and $\sigma_\theta = 0.2^o$. As shown in Figure 3(b), our rollout policy tends to select sensor B, which leads to a lower tracking error than that of CPA (see Figure 2(b)). Here, the CPA policy is the same as in scenario **1**, since sensor B has never been selected.

In our experiments, it is not surprising that the rollout policy outperforms CPA. What is significant here is that the rollout policy systematically and automatically trades off tracking performance and sensor-usage costs. In scenario **1**, we sacrifice some tracking performance for large reductions in sensor usage costs. In scenario **2**, we reduce the tracking error with no increase in sensor-usage costs.

6 Conclusion and Future Work

In this paper, we formulated the problem of sensor scheduling for target tracking as a POMDP, and proposed a general approach that combines particle filtering and Q-value approximation for solving the POMDP. As a particular instance of this approach, we implemented policy rollout with particle filtering. Our experiments on a simple sensor-scheduling problem involving multiple sensors for tracking a single target illustrates the effectiveness of this general approach.

Applying our approach to more complicated sensor-scheduling problems is part of our ongoing work. Specifically, we are currently investigating sensor scheduling problems with multiple targets and the selection of multiple sensors. For such multiple-target multiple-sensor scenarios, the state dynamics and observation map need to be extended accordingly. For the particle filter, we have some options: we can either construct a single particle filter for all targets or construct one particle filter for each target. The particle-filter algorithm and the Q-value approximation procedure remain the same, except with higher dimensions. The main additional feature needed in the multiple-target case is to incorporate a data-association module to decide which target is associated with each observation. Data association algorithms, such as JPDA (Joint Probabilistic Data Association), have been studied extensively. Our concern is simply to incorporate such algorithms into our approach. This turns out to be straightforward for the case of JPDA. For extensions of our approach involving the selection of multiple sensors, we also need to do sensor data fusion. This too is an area with an extensive literature from which to draw.

We can include time-varying and frequency-varying jamming sources into the state model by representing unobservable jamming intensities as additional state components. Similarly, we can incorporate measurements of jamming intensities into the observation model. This will enable us to deal with jamming, but will also impose additional costs. We expect these additional costs to be proportional to the number of jamming parameters.

A phenomenon known as "ghosting" is an important issue in multi-sensor arrays where two or more radar sensors, each limited in range resolution, interrogate an

environment containing two or more targets.[1] The ghosting phenomenon can cause a multi-sensor array to generate apparent target detections where there is no target. To deal with ghosting, a number of techniques have been proposed [18]. Typically, a "deghosting" system consists of an angle-only tracking filter, a triangulation range estimator, and a hypothesis test to determine if there are ghosts. Our method can handle ghosting by incorporating a particle filter to estimate true angles, angle velocities, and target ranges based on angle measurements, and perform hypothesis testing to eliminate ghosts.

Another direction for future research is to apply our work to a more general sensor-management problem, which includes sensor geometry control, sensor bandwidth allocation, sensor mode switching, as well as sensor scheduling. We may also consider integrating constraints into our POMDP formulation, such as battery capacity, load balancing, and bandwidth limits.

References

[1] S. Blackman, R. Popoli, Design and Analysis of Modern Tracking Systems, Artech House, Boston, 1999.

[2] Y. Oshman, Optimal sensor selection strategy for discrete-time state estimators, IEEE Transactions on Aerospace and Electronic Systems 30 (2) (1994) 307–314.

[3] A. Logothetis, A. Isaksson, On sensor scheduling via information theoretic criteria, in: Proceedings of the American Control Conference, San Diego, CA, USA, 1999, pp. 2402–2406.

[4] C. Kreucher, K. Kastella, A. O. Hero III, Multi-target sensor management using alpha-divergence measures, in: Proceedings of First IEEE Conference on Information Processing in Sensor Networks, Palo Alto, 2003.

[5] D. A. Castañon, Approximate dynamic programming for sensor management, in: Proceedings of the IEEE Conference on Decision and Control, San Diego, CA, USA, 1997, pp. 1202–1207.

[6] A. E. B. Lim, V. Krishnamurthy, Risk-sensitive sensor scheduling fo discrete-time nonlinear systems, in: Proceedings of the IEEE Conference on Decision and Control, Tampa, FL, USA, 1998, pp. 1859–1864.

[7] J. Evans, V. Krishnamuthy, Optimal sensor scheduling for hidden Markov model state estimation, International Journal of Control 74 (18) (2001) 1737–1742.

[8] V. Krishnamurthy, Algorithms for optimal scheduling of hidden markov model sensors, IEEE Transactions on Signal Processing 50 (6) (2002) 1382–1297.

[9] A. S. Chhetri, D. Morrell, A. Papandreou-Suppappola, Scheduling multiple sensors using particle filters in target tracking, in: Proceedings of 2003 IEEE Workshop Statistical Signal Processing,, 2003, pp. 549 – 552.

[1] We thank an anonymous reviewer for raising this issue.

[10] A. S. Chhetri, D. Morrell, A. Papandreou-Suppappola, The use of particle filtering with the unscented transform to schedule sensors multiple steps ahead, in: Proceedings of 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '04), Vol. 2, 2004, pp. 301–304.

[11] O. Hernandez-Lerma, Adaptive Markov Control Processes, Springer-Verlag, New York, 1980.

[12] D. Q. Mayne, H. Michalska, Receding horizon control of nonlinear systems, IEEE Transactions on Automatic Control 35 (7) (1990) 814–824.

[13] N. Gordon, B. Ristic, Tracking airborne targets occasionally hidden in the blind doppler, Digital Signal Processing 12 (2-3) (2002) 383–393.

[14] A. Doucet, N. de Freitas, G. Gordon, Sequential Monte Carlo Methods in Practice, Springer-Verlag, New York, 2001.

[15] C. Kwok, D. Fox, M. Meila, Real-time particle filters, Proceedings of The IEEE 92 (3) (2004) 469–484.

[16] D. P. Bertsekas, D. A. Castañon, Rollout algorithms for stochastic scheduling problems, Journal of Heuristics 5 (1999) 89–108.

[17] G. Wu, E. K. P. Chong, R. L. Givan, Burst-level congestion control using hindsight optimization, IEEE Transactions on Automatic Control 47 (6) (2002) 979–991.

[18] R. Yang, G. W. Ng, Deghosting in multi-passive acoustic sensors, in: Proceedings of SPIE - The International Society for Optical Engineering, Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications, Vol. 5434, 2004, pp. 187–194.

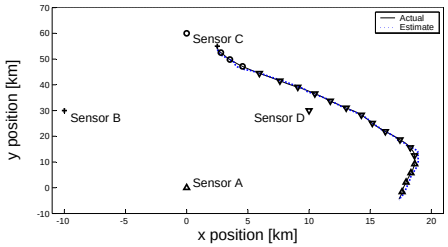
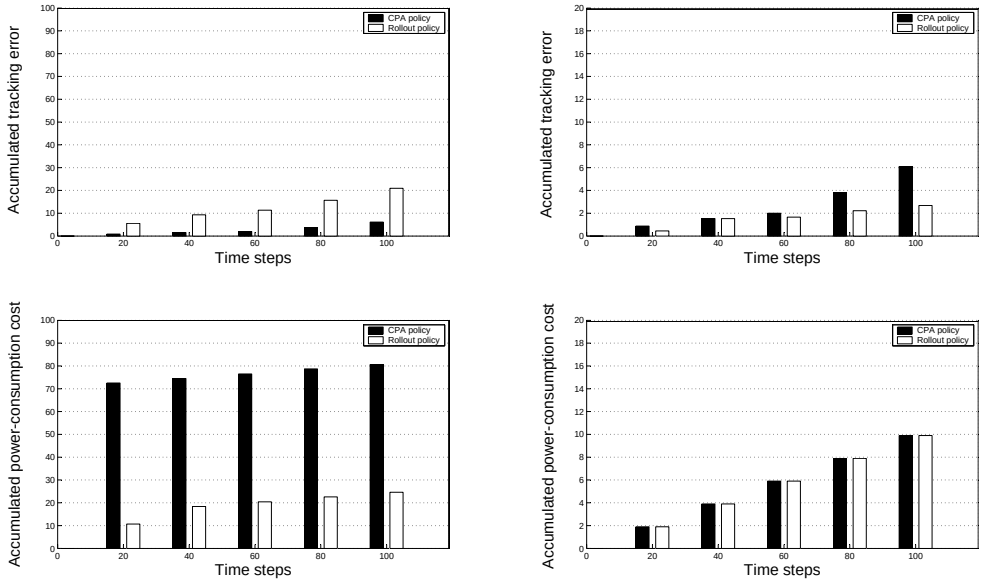
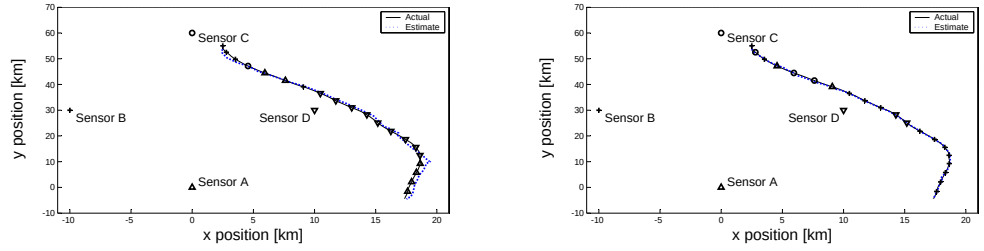


Fig. 1. Sensor selection and trajectory of the CPA policy (scenarios **1** and **2**).



(a) (b)

Fig. 2. Comparison of accumulated tracking errors and accumulated sensor power-consumption costs for the CPA and rollout policies. (a) scenario **1**; (b) scenario **2**.



(a) (b)

Fig. 3. Sensor selections and trajectories of the rollout policy (a) scenario **1**; (b) scenario **2**.

Binary Hypertree Classifiers for ATR

Definitions, Analysis, and Algorithms

Chad M. Spooner
Mission Research Corporation

March 5, 2004

Version 1.0

Abstract

The application of tree-based classifiers to automatic target recognition (ATR) and other classification problems is studied. The described work builds on a previous DARPA effort in which binary-tree classifiers were applied to ATR with range-doppler returns as input. The feature vectors required by the classifier are found during training by using an application of the local discriminant basis (LDB) wavelet-based technology. The extension of the LDB binary-tree ATR method to ISP is described in detail in this document. The fundamental idea is to connect a set of binary-tree classifiers in such a way that decisions at ambiguous nodes are resolved by requesting the best new measurement or statistic from the available sensor(s). In this way, the resulting binary hypertree classifier can have performance comparable to the supertree classifier that is informed of all sensor measurements but with an average input-data requirement that is substantially reduced by requesting only that data which is most relevant to the current point in the decision process.

Contents

List of Figures

1 Introduction

Automatic accurate determination of a radar target's type finds important application in tactical military situations. It may also have application in commercial aviation and search-and-rescue operations. More generally, automatic target recognition (ATR) is a specific example of an *automatic* M-*ary classification* problem [3]. Such problems can be found in many scientific and technological areas, such as type classification of RF signals, celestial objects, material compositions, tumors, heart diseases, bird calls, whale songs, etc.

In this report, we document progress on the development of a family of classification methods that integrate classification (processing) with measurement (sensing). The basic idea is to construct a classifier that first operates on a limited initial data set. If the classifier cannot produce a decision that has high quality (confidence), it requests the sensor measurement whose content will most decisively resolve the classification ambiguity. This iteration continues until a high-quality decision is reached or there are no further measurements available that can help reduce decision ambiguity.

Our specific approach relies on *binary-tree classifiers*. In particular, we build on the work we performed under the DARPA TRUMPETS program [2], in which we developed novel binary-tree classifiers for range-doppler inputs. The input-data statistics used to make the decision at each node are wavelet-based and the *local discriminant basis* (LDB) methodology [4] is adapted to find the best set of K such statistics independently for each node. This methodology can be viewed as a form of wavelet-based compression in which only the most discriminant portions of the data are retained, rather than those that yield the best reconstruction fidelity. In the present work, we develop an ISP classification framework in which collections of binary trees are joined together to form *binary hypertrees.* Each constituent binary tree in the hypertree is associated with a particular kind of sensor or sensor-target geometry. When an ambiguous node is reached during tree traversal for classification, the classifier jumps from the current tree to the tree that can best resolve the ambiguity. *This may necessitate obtaining a new sensor output.* Thus, the ideal is that the minimum amount of data is collected for the problem at hand and average performance for the hypertree is much better than for any single constituent tree.

The remainder of this document is organized as follows. The basic mathematical framework for ISP-enabled iterative classification (ATR) is presented in Section 2. This framework includes definitions of several tree-based classifiers. The concepts of operation for the various classifiers are described in Section 3 and an approximate performance analysis is provided in Sections 4 and 5. Algorithms for classifier construction and operation are provided in Section 6. An illustrative (toy) problem is examined in Section 7, and the applicability of the approach to a wide variety of classification problems and classifier structures is described in Section 8. Finally, conclusions are drawn in Section 9. In a later stage of this work, a companion report will provide detailed simulation results for the classifiers defined and described herein.

2 Mathematical Framework

In this section we present the basic mathematical definitions required to study the proposed tree-based classifiers. In subsequent sections we introduce several propositions and further definitions.

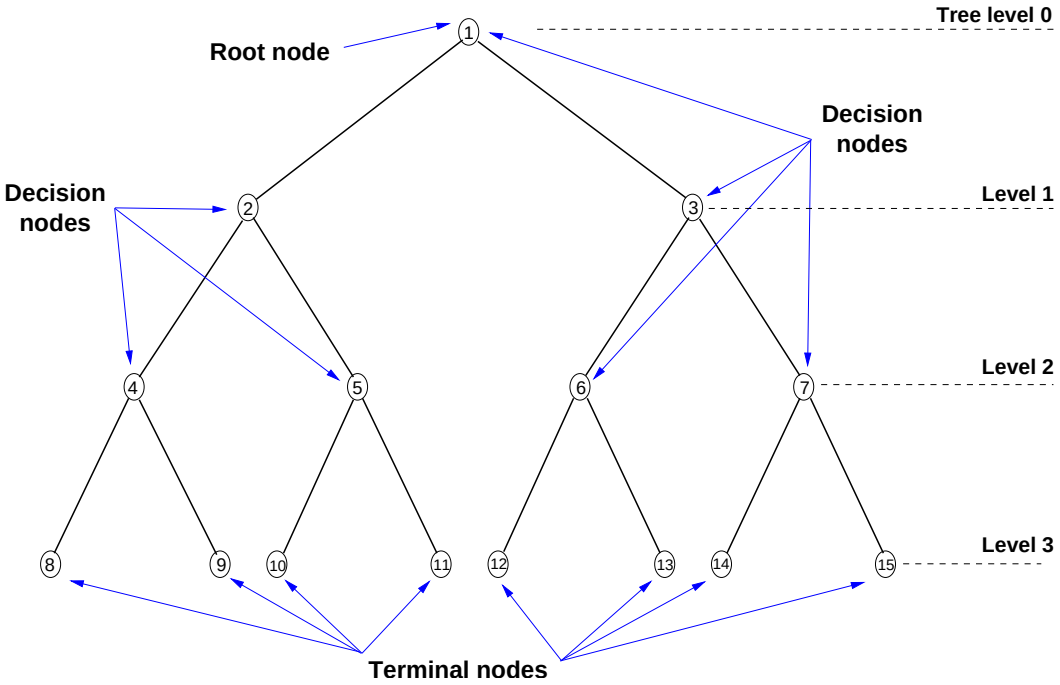


Figure 1: Graphical depiction of a generic binary tree for tree parameter $C = 8$.

Definition 1 (Classifier Input Data (CID)) *The CID quantity $\boldsymbol{X}$ is a vector or matrix that contains measurements obtained by sensing the environment in some manner and optionally processing this sensor data. $\boldsymbol{X}$ is then used as input to a classifier.* ■

Definition 2 (C-Class Problem) *Given class-membership labels $1, 2, \ldots, C$, and at least one CID $\boldsymbol{X}$, determine the class label $\hat{L}$ that corresponds to the CID, $\hat{L} = g(\boldsymbol{X})$, $\hat{L} \in \{1, 2, \ldots, C\}$. The function $g(\cdot)$ represents the classifier.* ■

Definition 3 (Binary Tree) *For the dyadic number C, a binary tree is a collection of $2C-1$ nodes with a specific node-connection topology. The root node is node number one and has two children. The final C nodes are childless. All other nodes have exactly one parent and two children. Each node is the child of only one parent. Node connections and numbering are as shown in Figure 1. The tree is said to be* balanced *if the number of nodes at level l is twice the number of nodes at level $l-1$ for each tree level greater than zero. Otherwise, the tree is* unbalanced. ■

Definition 4 (Decision Node) *A decision node in a binary tree with parameter C is any node except one of the final (childless) C nodes (see Figure 1).* ■

Definition 5 (Terminal Node) *A terminal node in a parameter-C binary tree is any node that is not a decision node.* ■

Definition 6 (Binary-Tree Classifier (BTC)) *A binary-tree classifier is a binary tree for parameter C such that each node is associated with a vector-valued measurement on the CID $\boldsymbol{X}$ and a binary decision between two mutually exclusive groups of class labels each called a superclass.*

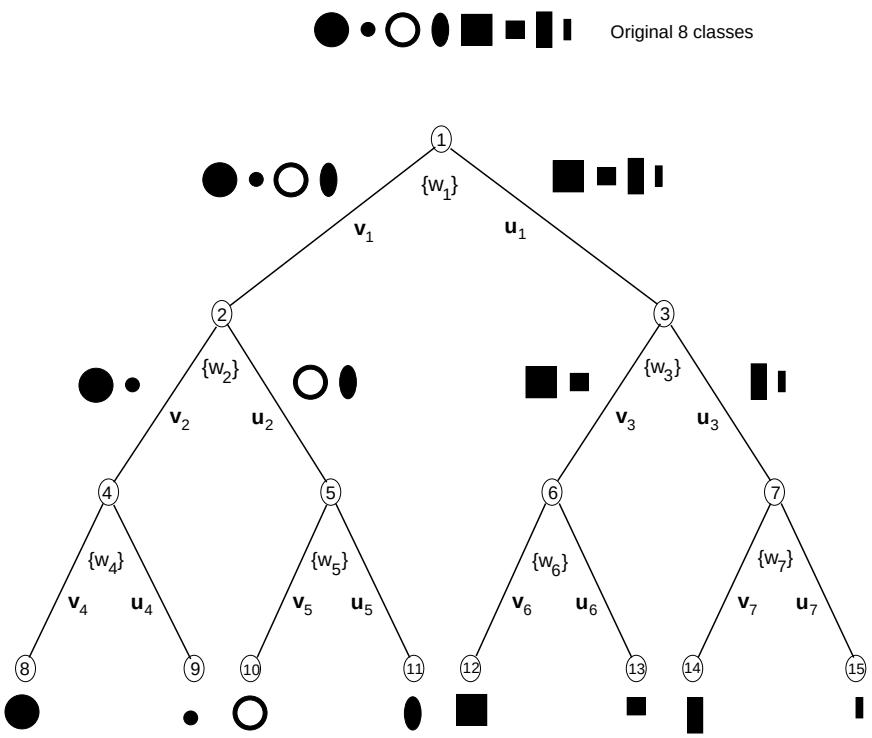


Figure 2: Graphical depiction of a generic balanced binary tree classifier for tree parameter (problem size) $C = 8$.

The superclasses for node i derive from splitting the inherited class label set from the parent of node i into two nonempty and disjoint sets. The average measurement vector value for the left class is denoted by $\boldsymbol{v}$ and that for the right by $\boldsymbol{u}$. The length of the measurement vector for all nodes is denoted by K. The BTC definition is illustrated by Figure 2. ■

Definition 7 (BTC Node Ambiguity) *For a BTC decision node n, the ambiguity A_n is defined by*

$$A_n = \frac{1}{2}(r_n + 1),$$

where

$$\begin{aligned} r_n &= \frac{\boldsymbol{u}_n \boldsymbol{v}_n^T}{\|\boldsymbol{u}_n\| \, \|\boldsymbol{v}_n\|}, \\ \|\boldsymbol{u}_n\| &= \left| \sum_{k=1}^{K} u_{n,k}^2 \right|^{1/2}. \end{aligned}$$

Therefore the node ambiguity is a simple function of the correlation coefficient (CC) r_n between the left and right feature vectors for node n. ■

Definition 8 (Node Descendents) *The descendents of binary-tree node n are all nodes that can be reached from n by traversing the tree starting at n. The descendents of the root node (node one) are all nodes other than node one. The set of descendents for a terminal node is empty.* ■

Definition 9 (Average Downbranch Ambiguity) *Let BTC node n have N descendents $\{n_i\}_{i=1}^N$. Then the average downbranch ambiguity (ADA) for node n is*

$$a_n = \frac{1}{N+1}(A_n + \sum_{k=1}^{N} A_{n_k}).$$

Therefore the total BTC ambiguity is a_1. ■

Definition 10 (Corresponding Nodes) *Let T_1 and T_2 denote two distinct BTCs and let n_1 and n_2 denote decision nodes from T_1 and T_2, respectively. If the union of the left and right superclasses for nodes n_1 and n_2 match, then these two nodes are* corresponding. *If the superclasses are equal, then the nodes are* equivalent. *Note that the binary-tree parameters C_1 and C_2 for T_1 and T_2 need not be equal.* ■

Definition 11 (Binary Hypertree) *Let $\{T_i\}_{i=1}^N$ denote a set of N binary trees with identical parameters C. Associate with each node n in each tree i an index in the set $\{1, 2, \ldots, N\}$ and a node index. Then this node n in tree i is said to contain a pointer to the indexed tree. The collection of binary trees so indexed is called a binary hypertree.* ■

Definition 12 (Binary Hypertree Classifier (BHC)) *Let H denote a binary hypertree with N constituent binary tree classifiers each with parameter C and each addressing the same classification problem. Assign the node pointers for each node such that the node points to the BTC with minimum ambiguity-corresponding node. To classify a CID, select an initial constituent BTC. Use the node pointers to switch constituent trees whenever a node does not point to itself. Upon switching to a BTC, if the CID for that BTC has not been obtained, direct the appropriate sensor to obtain the data and then proceed. Figure 3 depicts graphically a BHC for $N = 4$ and $C = 8$. Hypertree indices for decisions nodes 1, 2, and 4 are also shown in the figure.* ■

Definition 13 (Binary Supertree Classifier (BSC)) *A binary supertree classifier is a BTC for which the CID is multimodal. That is, the CID is a concatenated or otherwise stacked set of separate (distinct) CIDs, $Y = \{\mathbf{X}_j\}_{j=1}^M$. The BSC can be viewed as the binary-tree classifier that operates on all available sensor outputs.* ■

3 Concept of Operations

We will provide detailed algorithm statements for classifier construction and operation in Section 6. In this section we provide a high-level overview of the operation of the three main classifier types—BTC, BSC, and BHC—to provide sufficient context for the performance analysis of Sections 4 and 5.

Binary Tree Classifiers.
A typical binary tree classifier (BTC) is shown in Figure 2. This kind of classifier is constructed by obtaining a set of training data for one CID type and all classes and applying the LDB machinery of [2]. The resulting classifier must operate only on the CID type with which it was created. To operate the classifier, we begin in node one, compute the correlation between the measured feature

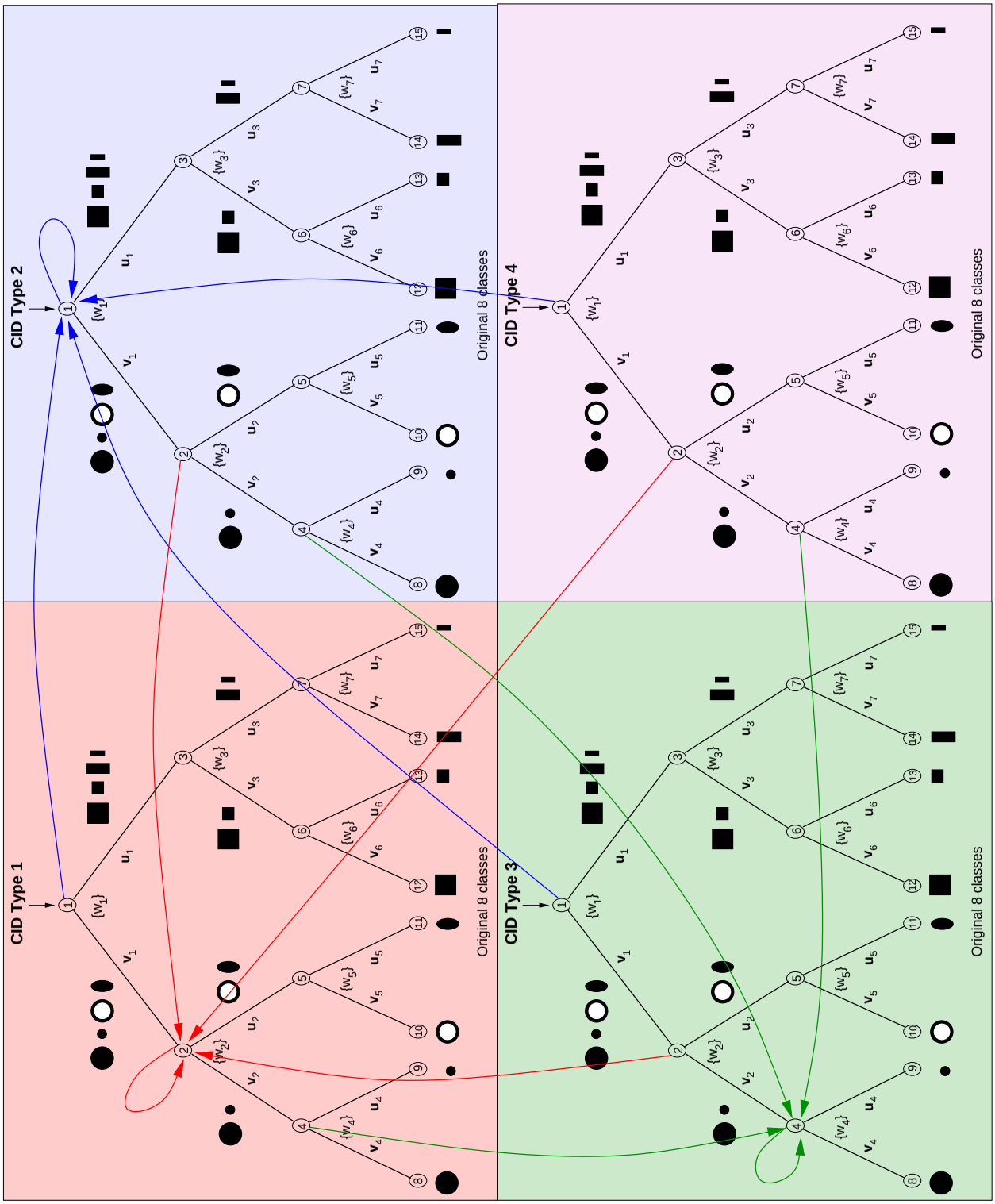


Figure 3: A binary hypertree illustration for an eight-class problem. The hypertree indices for nodes one, two, and four are shown.

and the left and right stored features and take the path out of the node indicated by the larger correlation. This is continued until a terminal node is reached.

Remark 1 *If the CID type is not sufficient to render the total tree ambiguity a_1 small, then average classification performance will likely be poor.*

Binary Supertree Classifiers.
A binary supertree classifier looks much like the BTC of Figure 2. The difference is that the input is a collection of CIDs of distinct types. The classifier is constructed by obtaining a set of training data for all CIDs of interest and all classes, and then applying the same LDB machinery as used for the BTC. If the set of CID types is exhaustive for a particular problem of interest (no more can be obtained from existing allocated resources), then the performance of the BSC sets an upper limit on the performance of the BTC.

Remark 2 *If the collection of employed CID types is not sufficient to render the total tree ambiguity a_1 small, then BSC performance will likely be poor. This means that the designated CIDs are inadequate for the problem at hand.*

Binary Hypertree Classifiers.
A binary hypertree classifier is shown in Figure 3 for $C = 8$ and $N = 4$. This kind of classifier is constructed by obtaining the same training data as used in the BSC, that is, M sets of CID data for each of the C classes, and then constructing at least one BTC for each of the M CID types. The BTCs are linked together as in Definition 12. To operate the classifier, choose an initial constituent BTC and obtain its CID. Traverse the BTC, jumping to another BTC whenever a decision node is encountered that does not point to itself. For every jump to another BTC, obtain the corresponding CID if it has not already been obtained during the traversal of the hypertree.

Remark 3 *The benefit of using a BHC is that BSC performance can in principle be achieved without the cost of routinely obtaining all M CID types for each traversal of the tree.*

4 Analysis for Balanced Trees

In this section we present some analysis results for the specific case of balanced trees (see Definition 3). Unbalanced trees are addressed in Section 5.

Each decision node in a BTC or BHC must make a binary decision based on a vector of K (wavelet) measurements on the CID. This decision is made by computing the correlation coefficients (CCs) between the measured feature vector $\boldsymbol{y}$ and the left and right superclass average feature vectors $\boldsymbol{v}$ and $\boldsymbol{u}$,

$$\begin{aligned} z_l &= CC(\boldsymbol{y}, \boldsymbol{v}) \\ z_r &= CC(\boldsymbol{y}, \boldsymbol{u}). \end{aligned}$$

If $z_l > z_r$, take the left decision path out of the node, else if $z_l < z_r$, take the right path, else flip a fair two-sided coin to determine which path to take.

If the CC between $\boldsymbol{v}$ and $\boldsymbol{u}$ is large and negative, the ambiguity will be close to zero. In this case, if the CID corresponds to a class in $\{1, 2, \ldots, C\}$ (i.e., it is represented by the tree, not an unknown class), then we expect the correct decision to be made at this node almost all of the time.

On the other hand, if the CC is large and positive, the ambiguity is close to one, and we expect the decision to be incorrect about half the time on average. Let us represent the node error probability as a function of the node ambiguity,

$$P_e(n) = f(A_n),$$

such that $f(0) = 0$ and $f(1) = 1/2$ and $f(\cdot)$ is continuous and monotonic on $[0, 1]$, bounded below by zero, and bounded above by one. An example is $f(x) = x/2$ for which we have $P_e(n) = A_n/2$.

4.1 Binary Tree Classifier

We now compute the BTC probability of correct classification for class $c \in \{1, 2, \ldots, C\}$. There is only one path through a BTC that terminates at class c. Let the node sequence that defines this path be denoted by $\{n_{c,1}, n_{c,2}, \ldots, n_{c,D}\}$, where $D = \log_2(C)$. This distinct path for class c is associated with the sequence of ambiguities $\{A_{n_{c,1}}, A_{n_{c,2}}, \ldots, A_{n_{c,D}}\}$.

Proposition 1 (Error Performance for a BTC)
Given a C-class problem and a BTC associated with that problem, assume that the decisions at each node in the path for class c are approximately independent. Then the average classification error for the BTC is given by

$$P_e(BTC) = \frac{1}{C}\sum_{c=1}^{C}\left[1 - \prod_{k=1}^{D}(1 - f(A_{n_{c,k}}))\right]. \quad (1)$$

■

Proof. The probability of correct classification, conditioned on the true input class of c, is the product of marginal probabilities of correct classification for each node in the path,

$$\begin{aligned} P(\hat{L} = c|c) &= \prod_{k=1}^{D} P_s(n_{c,k}) = \prod_{k=1}^{D}(1 - P_e(n_{c,k})) \\ &= \prod_{k=1}^{D}(1 - f(A_{n_{c,k}})). \end{aligned}$$

Thus, the conditional probability of error is given by

$$\begin{aligned} P_e(c) &= P(\hat{L} \neq c|c) \\ &= 1 - P(\hat{L} = c|c) \\ &= 1 - \prod_{k=1}^{D}(1 - f(A_{n_{c,k}})). \end{aligned}$$

If we assume that the prior probabilities for the C classes are equal, then the total probability of error is given by

$$\begin{aligned} P_e(BTC) &= \frac{1}{C}\sum_{c=1}^{C} P(\hat{L} \neq c|c) \\ &= \frac{1}{C}\sum_{c=1}^{C}\left[1 - \prod_{k=1}^{D}(1 - f(A_{n_{c,k}}))\right]. \end{aligned}$$

□

Note that the probability of error for the BTC is completely characterized by the node ambiguities. The limiting cases of completely ambiguous and unambiguous BTCs are treated in the next proposition.

Proposition 2 (BTC Error Probability Limiting Cases)
Given a BTC with parameter size C, if all node ambiguities are zero, then the probability of error for the BTC is zero. If all node ambiguities are equal to one, then the probability of error is equal to $(C-1)/C$. ■

For a BTC, the location in the tree of highly ambiguous nodes has a strong influence on performance. This is easy to see since the various decision nodes are components of different numbers of paths. If the root node is highly ambiguous with $A_1 = 1$, then all paths begin with an ambiguous decision and the probability of error for the tree is 1/2. The location of the ambiguous node is grossly characterized by its level in the tree (see Figure 1).

Proposition 3 (BTC Error as Function of Ambiguous Node Position)
Suppose we have a BTC with parameter C. Node n at tree level l has ambiguity $A_n = 1$ and all other decision nodes have zero ambiguity. Then the probability of error for the BTC is $2^{-(l+1)}$. ■

The average error can be minimized by ensuring that high-level nodes (small l) have minimum ambiguity. One way of doing this is by selecting the decision-node superclasses to push ambiguity downward in the tree. Other methods include extending the BTC to a BHC.

4.2 Binary Hypertree Classifier

For BHCs, each node in each of the constituent BTCs contains a pointer to the constituent BTC with minimum-ambiguity corresponding node. Therefore, the ambiguity $A_{n_{c,k}}$ seen at node $n_{c,k}$ for the path corresponding to class c in a BTC is replaced by

$$B_{n_{c,k}} = \min_{t \in T_{c,k}} A_{n_{c,k}}(t),$$

where t indexes the constituent BTCs with nodes corresponding to $n_{c,k}$.

Proposition 4 (Error Performance for a BHC)
The average error probability for a BHC is given by

$$P_e(BHC) = \frac{1}{C}\sum_{c=1}^{C}\left[1 - \prod_{k=1}^{D}(1 - f(B_{n_{c,k}}))\right].$$

■

The error performance for a BHC is completely characterized by the minimum-ambiguity corresponding nodes.

Error Bounds.

Let us now compute some bounds on the performance of a BHC. The general expression for the probability of error for the BHC is

$$P_e(BHC) = \frac{1}{C}\sum_{c=1}^{C} P_e(c),$$

where

$$P_e(c) = 1 - \prod_{k=1}^{D}(1 - f(B_{n_{c,k}})).$$

Now $0 \leq f(\cdot) \leq 1$ so that

$$0 \leq f(B_{n_{c,k}}) \leq 1$$

or

$$\prod_{k=1}^{D}(1 - f(B_{n_{c,k}})) \leq 1 - f(\beta_{n_c})$$

where

$$\beta_{n_c} = \arg\ \max_k f(B_{n_{c,k}}).$$

Therefore

$$\begin{aligned} -\prod_{k=1}^{D}(1 - f(B_{n_{c,k}})) &\geq -(1 - f(\beta_{n_c})) \\ P_e(c) &\geq 1 - (1 - f(\beta_{n_c})). \end{aligned}$$

Finally, then, the bound on the total error probability is given by

$$\begin{aligned} P_e(BHC) &\geq \frac{1}{C}\sum_{c=1}^{C}[1 - (1 - f(\beta_{n_c}))] \\ &= \frac{1}{C}\sum_{c=1}^{C} f(\beta_{n_c}) \\ &\geq f(\beta_n'), \end{aligned}$$

where $\beta_n' = \arg\ \max_c f(\beta_{n_c})$. So the average BHC error probability is bounded away from zero by the node with the largest minimized error probability over all constituent BTCs in the BHC.

When the error function $f(\cdot)$ is a nondecreasing function of the ambiguity, then $\beta_n' = \max_c \beta_{n,c}$. That is, the maximum error probability corresponds to the maximum ambiguity.

Remark 4 *The bound on error performance for the BHC suggests that to improve performance, find the most ambiguous node in the BHC paths and attempt to find a new CID, new vector-valued measurement on an existing CID, or new superclass definitions so as to result in a smaller ambiguity for this node. The minimum change to the BHC is then to have it point corresponding nodes to the new BTC.*

4.3 Binary Supertree Classifier

Suppose we have M distinct CID types available for use by our classification system. These could correspond to different radar waveforms (pulse widths), frequency bands, look directions, or sensor modality, such as optical, infrared, SAR, etc. A sample set of CIDs can be represented by the collection $\{\boldsymbol{X}_j\}_{j=1}^M$, where each $\boldsymbol{X}_j$ is a vector or matrix of data obtained from a particular active or passive sensing of the environment. A *binary supertree classifier* (BSC) is a binary tree classifier that takes as input this set of M CIDs (see Definition 13). Since the BSC is a BTC, its performance is characterized by Proposition 1.

The idea behind the BSC is that of a *maximally informed classifier*. Its performance serves as an upper limit on the performance of any other tree based classifier since it uses all available information in training and in operation.

Of course, the drawback of a BSC is that it might require an enormous amount of input data if the number of distinct CID types is large and the classification problem is difficult (different classes require distinct kinds of CIDs to achieve good performance). What we would like is the performance of the potentially impractical BSC with the relatively modest operational requirements of the BHC.

4.4 Relations Between Classifiers

In this section, we present analysis results pertaining to the relationships between the three tree-based classifier types. One of our aims is to determine the requirements on a BHC for exact correspondence between it and a BSC. Another aim is to determine error-performance requirements for the constituent BTCs of a BHC for good BHC performance: How good do the BTCs need to be to guarantee good BHC performance?

4.4.1 Relations Between BSCs and BHCs

From an intuitive point of view, the performance of a BSC should lower bound the performance of any BHC for the same problem because the BSC has all available CID types and can process them jointly at each decision node.

Let us first consider the class of BSCs for which the measurements made at each decision node require only one CID type; that is, each K-vector of measurements at node n requires only one element of the multimodal CID $Y = \{\mathbf{X}_j\}_{j=1}^M$. For this class of BSCs, we expect that the BSC performance can be exactly achieved by a relatively simple BHC. These considerations lead to the following two definitions.

Definition 14 (Reducible Binary Supertree Classifier) *Let the BSC S correspond to the multimodal CID $Y = \{\mathbf{X}_j\}_{j=1}^{M}$. S is* reducible *if the measurement vector for each decision node n is associated with only one element of Y. Otherwise the BSC is* irreducible. ■

Definition 15 (Simple Binary Hypertree Classifier) *Let the multimodal CID be represented by $Y = \{\mathbf{X}_j\}_{j=1}^{M}$. An associated BHC is called* simple *if all constituent BTCs have CIDs that correspond to a single element of Y. Otherwise, if at least one BTC has a multimodal CID, the BHC is called* complex. ■

Let's also formalize the notation for describing the required CID elements at a decision node.

Definition 16 (Mode Subset) *Let n represent a decision node of an irreducible BSC that is associated with the multimodal CID set $Y = \{\mathbf{X}_j\}_{j=1}^{M}$. The K-vector of measurements associated with n requires at least one element of Y and at most all M elements. Let $\mathbf{I}$ represent the vector of indices in $\{1, 2, \ldots, M\}$ that are actually required by node n. $\mathbf{I}$ is called the* mode subset vector. ■

The performance of any reducible BSC can be achieved by a simple BHC, which is the subject of the following proposition.

Proposition 5 (Equivalence Between Reducible BSC and Simple BHC) *Let S represent a reducible BSC associated with the C-class problem P_1 and the multimodal CID set $Y = \{\mathbf{X}_j\}_{j=1}^{M}$. Then there exists a simple BHC H for P_1 associated with Y such that $P_e(S) = P_e(H)$. Moreover, there are M BTCs in H.* ■

Proof. (By construction.) For each $j = 1, 2, \ldots, M$, construct a BTC T_j for $\{\mathbf{X}_j\}$ such that the superclasses for each node n in each BTC match those in S for node n. Therefore each decision node in the BSC has M corresponding nodes in the set of M BTCs. Consider decision node m in S. Since S is reducible, the feature vector for this node can be obtained from measurements on one of the CID types, say $\mathbf{X}_k$. For BTC T_k, associate the measurement specification, $\mathbf{u}$, and $\mathbf{v}$ for node m in S with node m in T_k. For all T_j, point node m to BTC T_k. Repeat this procedure for all decision nodes in S. The resulting set of M linked BTCs forms a BHC H. By construction, each possible traversal of S corresponds to an identical traversal of H, and there are no other traversals in either classifier. Therefore, $P_e(S) = P_e(H)$. □

Even when a BSC employs multimodal measurements at some or all decision nodes, its performance can still be obtained by using a properly designed BHC. If most or all of the decision nodes in the BSC use measurements that require most or all of the elements of Y, then there may be no operational benefit to using the equivalent BHC. But when only a few nodes require multimodal measurements, there can be great operational and training advantages to using the BHC over the BSC.

The relationship between irreducible BSCs and complex BHCs is summarized in the following proposition.

Proposition 6 (Equivalence Between Irreducible BSC and Complex BHC) *Let S represent an irreducible BSC associated with the multimodal CID set $Y = \{\mathbf{X}_j\}_{j=1}^{M}$ and a C-class problem P_1. Then there exists a complex BHC H associated with Y and P_1 such that $P_e(S) = P_e(H)$.* ■

Proof. (By construction.) First consider the decision nodes in S that have mode subset vectors with length one, say $\{q_j\}_{j=1}^{Q}$. For each q_j, construct a BTC Z_j such that its node q_j performs the same operations as S on the CID associated with q_j; point node q_j in Z_j to Z_j. For the remaining $Q-1$ BTCs, point their q_j nodes to Z_j. Now consider the decision nodes in S for which the mode subset vector has length greater than one, say, $\{p_j\}_{j=1}^{P}$. For each of these nodes, construct a new BTC in the following way. For node p_j, define a multimodal CID $Y_j = \{\mathbf{X}_k\}_{k\in\mathbf{I}_j}$, where $\mathbf{I}_j$ is the mode subset vector. Let Z_j' denote the BTC associated with p_j. The CID for Z_j' is Y_j, and for node p_j in Z_j', perform the same operations on Y_j as in S. Point the p_j nodes in all P new BTCs to Z_j'. If $Q > 0$, then examine new BTC Z_1 to determine how to point the single-mode nodes in the P new BTCs. Similarly, for the Q new single-mode BTCs, examine Z_1' to determine how to point the multimodal nodes. The resulting $P+Q$ BTCs define a complex BHC H since $P \geq 1$. By construction, each possible path through S has an identical path through H. Therefore, $P_e(S) = P_e(H)$. □

Remark 5 *For BSCs that are only mildly irreducible (meaning that $P \ll Q$), the number of constituent BTCs in the BHC will be only slightly larger than M, yet the performance will be equal to that of the BSC.*

4.4.2 Approximation of BHCs by BHCs

We have demonstrated that any BSC can be exactly represented by a BHC. Since the computational and storage burden of the BHC is directly related to the number of constituent BTCs and their CIDs (multimodal or single-mode), we are now interested in the possibility of approximating complicated BHCs with simpler ones. So we investigate the performance penalty in approximating one BHC by another.

Incremental Approximation.

Let us first consider the smallest possible difference between two BHCs: a difference in a single decision node. For example, suppose that in BHC H_1, node n obtains its minimum ambiguity for constituent BTC k, which has a multimodal CID. Let the dimension of the mode subset vector for node n in BTC k be $v > 1$. Consider now another BTC with the same superclasses as k but for which the CID has dimension $v-1$. Form a new BHC, H_2, for which all corresponding nodes with indices n point to this new, lower-dimensional BTC. All other corresponding nodes (those with indices other than n) in all trees of H_2 take the same values as their counterparts in H_1. Thus, the only difference between H_1 and H_2 occurs for decision node n.

Proposition 7 (Incremental BHC Error Performance)
Let H_1 and H_2 be BHCs for a given C-class problem and multimodal CID $Y = \{\mathbf{X}_j\}_{j=1}^{M}$. Let H_1 and H_2 differ only in a single decision node n for which the minimum ambiguity over all corresponding nodes with index n is larger in H_2 than in H_1. Then the performance difference for the two BHCs is given by

$$P_e(H_2) - P_e(H_1) \approx \frac{f(B_n') - f(B_n)}{2^{k_0-1}},$$

where B_n' and B_n are the minimum ambiguities for H_2 and H_1, respectively, at node n, and $k_0 - 1$ denotes the tree level of node n. ■

Proof. Recall that the performance of a BHC is given by

$$P_e(H) = \frac{1}{C}\sum_{c=1}^{C}\left[1 - \prod_{k=1}^{D}(1 - f(B_{n_{c,k}}))\right]$$

where $D = \log_2(C)$, $B_{n_{c,k}}$ is the minimum-ambiguity node over all constituent BTCs for node $n_{c,k}$, which is the kth node encountered along the unique path to class c. The performance difference is computed straightforwardly as follows

$$\begin{aligned} P_e(H_2) - P_e(H_1) &= \frac{1}{C}\sum_{c=1}^{C}\left[1 - \prod_{k=1}^{D}(1 - f(B'_{n_{c,k}}))\right] - \frac{1}{C}\sum_{c=1}^{C}\left[1 - \prod_{k=1}^{D}(1 - f(B_{n_{c,k}}))\right] \\ &= \frac{1}{C}\sum_{c=1}^{C}\left[-\prod_{k=1}^{D}(1 - f(B'_{n_{c,k}})) + \prod_{k=1}^{D}(1 - f(B_{n_{c,k}}))\right] \\ &= \frac{1}{C}\sum_{c=1}^{C}\left(f(B'_{n_{c,k_0}}) - f(B_{n_{c,k_0}})\right)\prod_{k\neq k_0}(1 - f(B_{n_{c,k}})) \end{aligned}$$

Since the node in question is at level $k_0 - 1$, it is part of exactly $C/2^{k_0-1}$ paths. For all other paths through the BHC, the difference between H_1 and H_2 is zero. Therefore, the performance difference is given by

$$P_e(H_2) - P_e(H_1) = \frac{1}{C}\sum_{c\in U}(f(B'_n) - f(B_n))\prod_{k\neq k_0}(1 - f(B_{n_{c,k}})),$$

where U denotes the set of class indices for which the unique path from root node to terminal node includes node n. If the BHC H_1 is sufficiently well constructed and the CID is sufficient for good classification performance, then we make the approximation

$$\prod_{k\neq k_0}(1 - f(B_{n_{c,k}})) \approx 1,$$

which implies that

$$P_e(H_2) - P_e(H_1) \approx \frac{f(B'_n) - f(B_n)}{2^{k_0-1}}.$$

□

Remark 6 *Proposition 7 implies that low-dimensional node approximations are better suited, in general, to parts of the tree that are nearer the terminal nodes (larger k_0).*

Now let's look at a more general approximation of a BHC by a simpler, lower-dimensional BHC. For BHC H_1, the minimum decision node ambiguities are denoted by $B_{n_{c,k}}$ and for BHC H_2, which approximates H_1, the ambiguities are denoted by $B'_{n_{c,k}}$.

Proposition 8 (Approximation of a BHC)
Let the BHCs H_1 and H_2 correspond to the same C-class problem and multimodal CID $Y = \{\mathbf{X}_j\}_{j=1}^{M}$. H_2 approximates H_1 by employing lower-dimensional CIDs at one or more nodes, or

by using a smaller value of K at one or more nodes. Assume that the minimum decision-node ambiguities for H_1 lower bound those for H_2: $B'_{n_{c,k}} \geq B_{n_{c,k}}$. Then

$$P_e(H_2) - P_e(H_1) \approx \frac{1}{C}\sum_{c=1}^{C}\left[\sum_{l=1}^{D}(f(B'_{n_{c,k}}) - f(B_{n_{c,k}}))\prod_{k=1,k\neq l}^{D}(1 - f(B_{n_{c,k}}))\right].$$

■

Proof. From the proof of Proposition 7, the performance difference between H_2 and H_1 can be expressed as

$$P_e(H_2) - P_e(H_1) = \frac{1}{C}\sum_{c=1}^{C}\left[\prod_{k=1}^{D}(1 - f(B_{n_{c,k}})) - \prod_{k=1}^{D}(1 - f(B'_{n_{c,k}}))\right].$$

By the monotonicity of $f(\cdot)$, we have

$$f(B'_{n_{c,k}}) \geq f(B_{n_{c,k}}) \ \ \forall c, k,$$

that is, the error probabilities for H_2 are no smaller than those for H_1. It is convenient to represent the errors for H_2 in terms of those for H_1 plus an additional term,

$$f(B'_{n_{c,k}}) = f(B_{n_{c,k}}) + e_{n_{c,k}},$$

where $e_{n_{c,k}} \geq 0$. Then the error difference can be represented by

$$P_e(H_2) - P_e(H_1) = \frac{1}{C}\sum_{c=1}^{C}\prod_{k=1}^{D}(1 - f(B_{n_{c,k}})) - \prod_{k=1}^{D}(1 - f(B_{n_{c,k}}) - e_{n_{c,k}}).$$

By computing the product involving $e_{n_{c,k}}$ and retaining only terms that are independent of e or are linear in e, we obtain

$$P_e(H_2) - P_e(H_1) \approx \frac{1}{C}\sum_{c=1}^{C}\left(\sum_{l=1}^{D} e_{n_{c,k}}\prod_{k\neq l}^{D}(1 - f(B_{n_{c,k}}))\right),$$

which is the desired result. □

4.4.3 Relations Between BHC and its Constituent BTCs

In this section, we consider the requirements on the constituent BTCs of a BHC for good BHC performance. Since the BHC performance depends only on the minimum-ambiguity corresponding decision nodes, it appears that excellent BHC performance may be had as long as there is a BTC with low ambiguity for each decision node. The BTC nodes that do not achieve the minimum ambiguity are irrelevant to BHC performance and can be highly ambiguous. These considerations suggest the definition of a peculiar kind of BTC that is good at making only a single decision.

Definition 17 (Savant Binary Tree Classifier) *A BTC for a C-class problem is* ϵ-savant *if one of its decision nodes has ambiguity less than or equal to ϵ and all other decision nodes have ambiguity greater than or equal to $1 - \epsilon$.* ■

The performance for a savant BTC depends strongly on the position of the unambiguous node in the tree, as shown in the following proposition.

Proposition 9 (Performance for a Savant BTC)
Suppose we have a savant BTC T with parameter ϵ and the unambiguous node is at tree level $k_0 \in \{0, 1, \ldots, D-1\}$. Then the performance for T is given by

$$P_e(T) \approx \frac{1}{C2^{k_0}}(2^{k_0}(C-1) - 1 + 2\epsilon).$$

provided that $f(\epsilon) \approx \epsilon$ and $f(1-\epsilon) \approx 1/2$. ■

Proof. The low-ambiguity node for T must be part of exactly $C/2^{k_0}$ paths through the tree. From (1), we can approximate the performance for T as follows

$$P_e(T) \approx \frac{1}{C}\left[\sum_{c\in U_1}\left(1 - \prod_{k=1}^{D}(1-f(1-\epsilon))\right) + \sum_{c\in U_2}\left(1-(1-f(\epsilon))\prod_{k\neq k_0+1}^{D}(1-f(1-\epsilon))\right)\right],$$

where $|U_1| = C - C/2^{k_0}$ and $|U_2| = C/2^{k_0}$. For small ϵ, $f(1-\epsilon) \approx 1/2$ and $f(\epsilon) \approx \epsilon$ so that

$$P_e(T) \approx \frac{1}{C}\left[(C - C/2^{k_0})(1 - (\frac{1}{2})^D) + C/2^{k_0}(1 - (1-\epsilon)(\frac{1}{2})^{D-1})\right].$$

After further simplification, we arrive at

$$P_e(T) \approx \frac{1}{C2^{k_0}}[2^{k_0}(C-1) - 1 + 2\epsilon].$$

□

Remark 7 *The limiting cases for the performance of a savant BTC correspond to $k_0 = 0$ and $k_0 = D-1$. For the former, the small-ambiguity node is the root node and we have $P_e \approx (C-2)/C$, which is slightly better than the worst case in which all nodes have high ambiguity. For the latter case, the small-ambiguity node is just above the layer of terminal nodes and $P_e \approx (C-1)/C - 2/C^2$ which is, again, slightly better than the worst case.*

Now if a BHC is made up of savant BTCs and every decision node in the BHC is covered by the unambiguous node for at least one of the BTCs, then we expect BHC performance to be very good independently of the poor performance of the BTCs. This leads to the definition of a savant covering.

Definition 18 (Savant Covering) *For a C-class problem, consider the collection of N distinct BTCs $\{T_k\}_{k=1}^N$. If for each set of corresponding nodes that are defined by this collection, there is at least one BTC with node ambiguity less than or equal to ϵ, then the BTC set is called a* covering. *If in addition the N BTCs are each ϵ-savant, then the covering is an* ϵ-savant covering. ■

Proposition 10 (BTC and BHC Performance for Savant Covering)
Let H represent a BHC for a given C-class problem and let $\{T_k\}_{k=1}^N$ represent the constituent BTCs for H. If the BTCs form a covering with parameter ϵ then the performance for H obeys the following bound

$$P_e(H) \leq 1 - (1 - f(\epsilon))^D.$$

Moreover, if the covering is ϵ-savant, then the performances for the constituent BTCs are bounded by

$$P_e(T_k) \geq \frac{C-2}{C}, \quad \forall k.$$

■

Proof. Since the constituent BTCs form a cover with parameter ϵ, the minimum ambiguity for any set of corresponding nodes is less than or equal to ϵ. Therefore, performance for the BHC is given by

$$P_e(H) = \frac{1}{C}\sum_{c=1}^{C}\left[1 - \prod_{k=1}^{D}(1 - f(B_{n_{c,k}}))\right],$$

where each $f(B_{n_{c,k}}) \leq f(\epsilon)$. Thus, we can obtain the desired result in the following way

$$\begin{aligned}
-f(B_{n_{c,k}}) &\geq -f(\epsilon) \\
1 - f(B_{n_{c,k}}) &\geq 1 - f(\epsilon) \\
\prod_{k=1}^{D}(1 - f(B_{n_{c,k}})) &\geq \prod_{k=1}^{D}(1 - f(\epsilon)) = (1 - f(\epsilon))^D \\
-\prod_{k=1}^{D}(1 - f(B_{n_{c,k}})) &\leq -(1 - f(\epsilon))^D \\
\frac{1}{C}\sum_{c=1}^{C}[1 - \prod_{k=1}^{D}(1 - f(B_{n_{c,k}}))] &\leq \frac{1}{C}\sum_{c=1}^{C}[1 - (1 - f(\epsilon))^D] \\
P_e(H) &\leq 1 - (1 - f(\epsilon))^D.
\end{aligned}$$

We have established the performance bound for the hypertree classifier for any ϵ covering. If the BTCs form an ϵ-savant covering, then

$$P_e(T_k) = \frac{1}{C2^{k_0(k)}}\left[2^{k_0(k)}(C-1) - 1 + 2\epsilon\right],$$

where $k_0(k)$ is the level of the ϵ-ambiguity node in tree T_k. The two performance extremes correspond to $k_0(k) = 0$ and $D-1$. Since the performance for $k_0(k) = 0$ is better than that for any other $k_0(k)$, we have

$$P_e(T_k) \geq \frac{1}{C}[C - 2 + 2\epsilon],$$

thereby establishing the proposition. □

Remark 8 *Proposition 10 implies that a high-performance classifier can be constructed from a set of classifiers with very poor performances provided that these classifiers can each make a single low-error decision.*

For the case in which constituent BTCs are not savant, we have the following proposition.

Proposition 11 (Bounded BTC Node Ambiguities)
Let the BHC H have constituent BTCs $\{T_k\}_{k=1}^N$. If the maximum node ambiguity for each T_k is no larger than $\epsilon = f^{-1}[1-(1-\delta)^{1/D}]$, where $D = \log_2(C)$, then the performance for H obeys $P_e(H) \leq \delta$. ■

Proof. It follows easily from the premises that the performance for H is bounded by

$$P_e(H) \leq 1 - (1 - f(\epsilon))^D.$$

Setting δ equal to the right side of this inequality yields the solution for ϵ,

$$\epsilon = f^{-1}(1 - (1-\delta)^{1/D}).$$

□

5 Analysis for Unbalanced Trees

What are the expected conceptual differences between the performances of balanced and unbalanced tree classifiers? Because the unbalanced trees have more degrees of freedom in superclass selection, one would think that the performance for the best unbalanced tree would lower bound the performance for the best balanced tree.

In this section, we revisit the propositions of the previous section on balanced trees to determine the major mathematical differences between the two tree types.

Like balanced trees, we impose the constraint that the superclasses at each decision node be mutually exclusive. Unlike balanced trees, the sizes of the superclasses at a node do not have to be equal. Therefore, there is still only a single path through the tree from root node to terminal node for each class. The length of each path is now a function of the class label c. For example, consider the unbalanced tree in Figure 4. The path lengths are denoted by $D(c)$ and range from 2 to 5.

As before, we assume that the successive decisions made at the decision nodes are approximately independent. The sequence of nodes visited as the tree is traversed along the unique path from root node to terminal node for class c is, as before, $\{n_{c,k}\}$.

Proposition 12 (Unbalanced BTC Error Performance)
The average error performance for an unbalanced BTC T that corresponds to a C-class problem is given by

$$P_e(T) = \frac{1}{C}\sum_{c=1}^{C}\left[1 - \prod_{k=1}^{D(c)}(1 - f(A_{n_{c,k}}))\right].$$

Proof. The proof is similar to that for Proposition 1. □

Next we revisit the limiting cases of the basic BTC performance formula.

Proposition 13 (Unbalanced BTC Error Probability Limiting Cases)
Given an unbalanced BTC with parameter size C, if all node ambiguities are zero, then the probability of error for the BTC is zero. If all node ambiguities are equal to one, then the probability of error is equal to $(C-1)/C$. ■

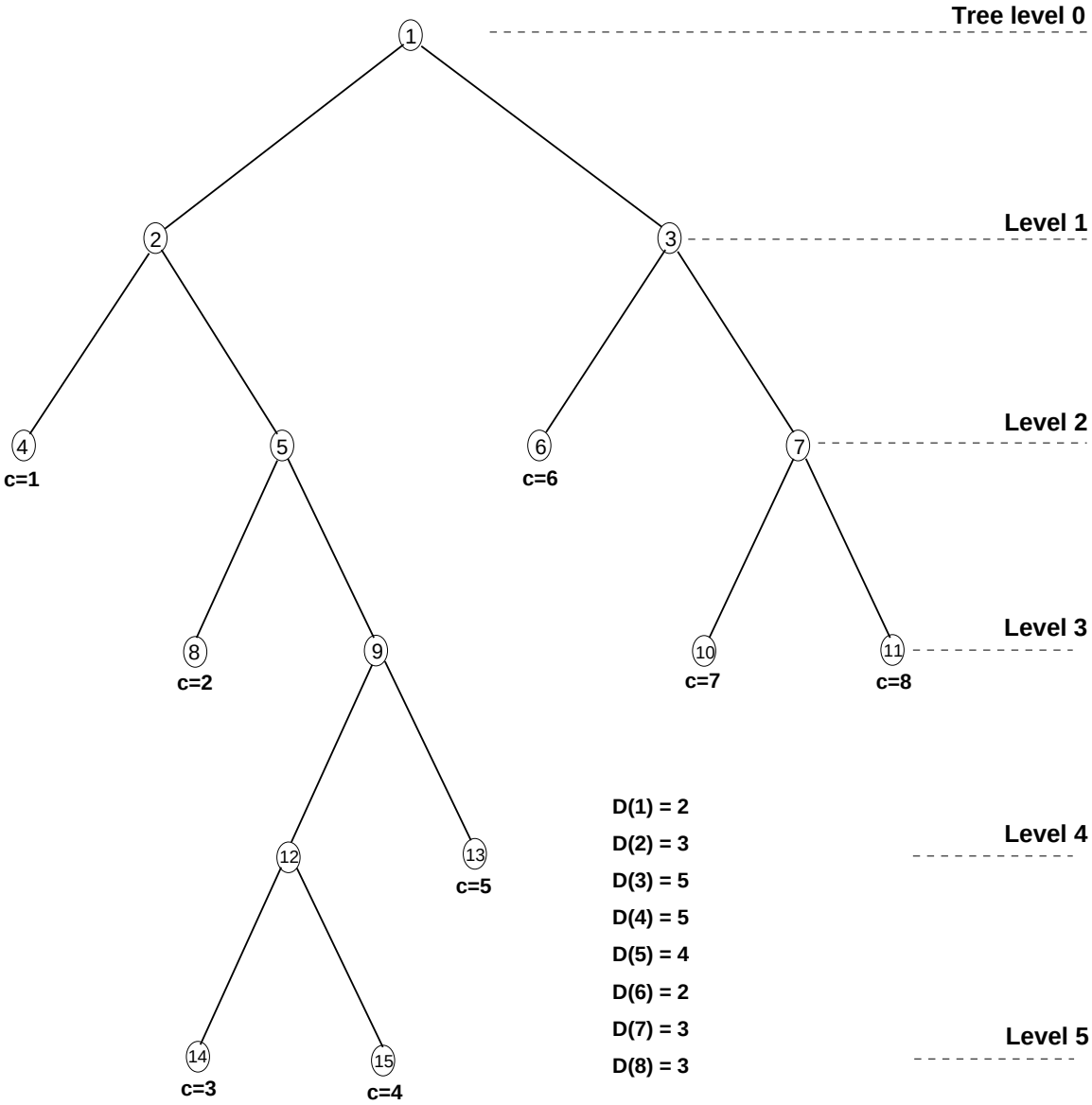


Figure 4: Illustration of the variable tree-traversal path length in an unbalanced tree classifier.

Proof. TBD. □

Thus, the limiting cases for balanced or unbalanced BTCs are the same. We now turn to the more interesting problem of determining BTC performance as a function of the position of a single ambiguous node. Our general expression for the probability of error is given by

$$P_e(T) = \frac{1}{C}\sum_{c=1}^{C}\left[1 - \prod_{k=1}^{D(c)}(1 - f(A_{n_{c,k}}))\right].$$

From Figure 4, we see that the tree level of an ambiguous node does not uniquely determine how many class paths are influenced by the node. Thus, we cannot find a formula for performance that depends only on the node's level. However, intuitively, the number of class paths is still important, which leads to the following definition.

Definition 19 (Node Impact) *The* impact *of a decision node n in an arbitrary BTC is the number of terminal nodes in n's descendents. Thus the impact must be an integer between two and C.*

Proposition 14 (Unbalanced BTC Error as a Function of Ambiguous Node Position) *For a C-class problem, let the unbalanced BTC T have a single ambiguous node n with ambiguity $A_n = 1$ and impact I. Then the error performance for T is*

$$P_e(T) = \frac{I_n}{2C}.$$

Proof. The generic BTC performance formula is

$$P_e(T) = \frac{1}{C}\sum_{c=1}^{C}\left[1 - \prod_{k=1}^{D(c)}(1 - f(A_{n_{c,k}}))\right].$$

Denote by C_I the subset of class labels for which node n lies along the path leading to the label. For all c not in C_I, the contribution to the error is zero since all nodes have zero ambiguity for these paths. Thus, the performance is given by

$$P_e(T) = \frac{1}{C}\sum_{c\in C_I}\left[1 - \prod_{k=1}^{D(c)}(1 - f(A_{n_{c,k}}))\right].$$

Assuming that $f(1) = 1/2$ yields

$$\begin{aligned} P_e(T) &= \frac{1}{C}\sum_{c\in C_I}\left[1 - (1-1/2)(1)^{D(c)-1}\right] \\ &= \frac{I}{2C}. \end{aligned}$$

□

Remark 9 *Note that the performance for an unbalanced BTC reduces to that for the balanced case because the impact must be of the form $I = C/2^{k_0}$. Thus $P_e = I/2C = (C/2^{k_0})/(2C) = 2^{-(k_0+1)}$.*

Proposition 15 (Error Performance for an Unbalanced BHC)
The average error probability for an unbalanced BHC is given by

$$P_e(BHC) = \frac{1}{C}\sum_{c=1}^{C}\left[1 - \prod_{k=1}^{D}(c)(1 - f(B_{n_{c,k}}))\right].$$

■

Proof. Similar to that for Proposition 4. □

Proposition 16 (Reducible BSC and Simple BHC for Unbalanced Trees)
Let S represent a reducible unbalanced BSC associated with the C-class problem P_1 and the multimodal CID set $Y = \{\mathbf{X}_j\}_{j=1}^{M}$. Then there exists a simple unbalanced BHC H for P_1 associated with Y such that $P_e(S) = P_e(H)$. Moreover, there are M BTCs in H. ■

Proof. Similar to that for Proposition 5. □

Proposition 17 (Irreducible BSC and Complex BHC for Unbalanced Trees)
Let S represent an irreducible unbalanced BSC associated with the multimodal CID set $Y = \{\mathbf{X}_j\}_{j=1}^{M}$ and a C-class problem P_1. Then there exists a complex unbalanced BHC H associated with Y and P_1 such that $P_e(S) = P_e(H)$. ■

Proof. Similar to that for Proposition 6. □

Proposition 18 (Incremental Unbalanced BHC Error Performance)
Let H_1 and H_2 be unbalanced BHCs for a given C-class problem and multimodal CID $Y = \{\mathbf{X}_j\}_{j=1}^{M}$. Let H_1 and H_2 differ only in a single decision node n for which the minimum ambiguity over all corresponding nodes with index n is larger in H_2 than in H_1. Then the performance difference for the two BHCs is given by

$$P_e(H_2) - P_e(H_1) \approx \frac{I}{C}(f(B'_n) - f(B_n)),$$

where B'_n and B_n are the minimum ambiguities for H_2 and H_1, respectively, at node n, and I denotes the impact of node n. ■

Proof. Similar to that for Proposition 7, except the node impact determines the underlying performance of the BTCs and BHC. □

Proposition 19 (Approximation of an Unbalanced BHC)
Let the unbalanced BHCs H_1 and H_2 correspond to the same C-class problem and multimodal CID $Y = \{\mathbf{X}_j\}_{j=1}^{M}$. H_2 approximates H_1 by employing lower-dimensional CIDs at one or more nodes, or by using a smaller value of K at one or more nodes. Assume that the minimum decision-node ambiguities for H_1 lower bound those for H_2: $B'_{n_{c,k}} \geq B_{n_{c,k}}$. Then

$$P_e(H_2) - P_e(H_1) \approx \frac{1}{C}\sum_{c=1}^{C}\left[\sum_{l=1}^{D(c)}(f(B'_{n_{c,k}}) - f(B_{n_{c,k}}))\prod_{k=1,k\neq l}^{D(c)}(1 - f(B_{n_{c,k}}))\right].$$

■

Proof. Similar to that for Proposition 8, except the sums over c have variable limits set by $D(c)$. □

Proposition 20 (Performance for an Unbalanced Savant BTC)
Suppose we have an unbalanced savant BTC T with parameter ϵ and the unambiguous node has impact I. Let the set of class labels which have tree-traversal paths that do not contain the unambiguous node be denoted by U_1 and the rest by U_2. Then the performance for T is given by

$$P_e(T) \approx \frac{1}{C}\left[C - \sum_{c\in U_1} 2^{-D(c)} - 2(1-\epsilon)\sum_{c\in U_2} 2^{-D(c)}\right]$$

provided that $f(\epsilon) \approx \epsilon$ and $f(1-\epsilon) \approx 1/2$. ■

Proof. For any BTC, the performance formula is given by

$$P_e(T) = \frac{1}{C}\sum_{c=1}^{C}\left[1 - \prod_{k=1}^{D(c)}(1 - f(A_{n_{c,k}}))\right].$$

We can divide the sum over c into two components, one which contains a contribution from the unambiguous node, and one which does not,

$$P_e(T) = \frac{1}{C}\left[\sum_{c\in U_1}(1 - \prod_{k=1}^{D(c)}(1 - f(1-\epsilon))) + \sum_{c\in U_2}(1 - (1 - f(\epsilon))\prod_{k\neq k_c}(1 - f(1-\epsilon))),\right]$$

where k_c denotes the index of the unambiguous node in the path $\{n_{c,k}\}$ for class c. Note that $|U_2| = I$, and $|U_1| = C - I$. Straightforward algebra leads to the desired result. □

Proposition 21 (Unbalanced BTC and BHC Performance for Savant Covering)
Let H represent an unbalanced BHC for a given C-class problem and let $\{T_k\}_{k=1}^{N}$ represent the constituent BTCs for H. If the BTCs form a covering with parameter ϵ then the performance for H obeys the following bound

$$P_e(H) \leq 1 - (1 - f(\epsilon))_*^D,$$

where $D_* = \max_c D(c)$. ■

Proof. By the definition of a cover, we have the fundamental inequality for all decision nodes in the BHC $B_{n_{c,k}} \leq \epsilon$ which, by the monotonicity of $f(\cdot)$, yields $f(B_{n_{c,k}}) \leq f(\epsilon)$. We obtain the following sequence of inequalities

$$\begin{aligned}
-f(B_{n_{c,k}}) &\geq -f(\epsilon)\\
1 - f(B_{n_{c,k}}) &\geq 1 - f(\epsilon)\\
\prod_{k=1}^{D(c)}(1 - f(B_{n_{c,k}})) &\geq \prod_{k=1}^{D(c)}(1 - f(\epsilon))\\
\prod_{k=1}^{D(c)}(1 - f(B_{n_{c,k}})) &\geq (1 - f(\epsilon))^{D(c)}
\end{aligned}$$

$$
\begin{aligned}
-\prod_{k=1}^{D(c)}(1-f(B_{n_{c,k}})) &\leq -(1-f(\epsilon))^{D(c)} \\
1-\prod_{k=1}^{D(c)}(1-f(B_{n_{c,k}})) &\leq 1-(1-f(\epsilon))^{D(c)} \\
\frac{1}{C}\sum_{c=1}^{C}1-\prod_{k=1}^{D(c)}(1-f(B_{n_{c,k}})) &\leq \frac{1}{C}\sum_{c=1}^{C}1-(1-f(\epsilon))^{D(c)} \\
P_e(H) &\leq \frac{1}{C}\sum_{c=1}^{C}1-(1-f(\epsilon))^{D(c)}.
\end{aligned}
$$

Since $f(\epsilon)$ must lie between zero and one, we have

$$(1-f(\epsilon))^{D(c)} \geq (1-f(\epsilon))^{D_*}$$

and therefore that

$$P_e(H) \leq \frac{1}{C}\sum_{c=1}^{C}1-(1-f(\epsilon))^{D_*}.$$

□

Proposition 22 (Bounded Unbalanced BTC Node Ambiguities)
Let the unbalanced BHC H have constituent BTCs $\{T_k\}_{k=1}^{N}$. If the maximum node ambiguity for each T_k is no larger than $\epsilon = f^{-1}[1-(1-\delta)^{1/D_*}]$, where $D_* = \max_c D(c)$, then the performance for H obeys $P_e(H) \leq \delta$. ■

Proof. The proof is similar to that for Proposition 11. □

6 Algorithms

In this section, we provide high-level algorithm statements for construction and use of the tree-based classifiers. Construction consists of exploiting the LDB to automatically find the most discriminating statistics in the training set. Classifier use simply means using the classifier to classify a CID. We begin with some notation so that the algorithm statements are sufficiently concise.

6.1 Notation

We adopt a programming-style notation consisting of variables and their records. Let btc denote a single binary tree classifier (BTC) and bhc denote a BHC. These trees are associated with the following records.

btc.cidType A string or numeric indicating the type of input used to train the BTC. For example, "range-doppler chips" or "optical image."

btc.angle The viewing angle for the targets (equivalent to pose).

btc.C The size of the classification problem addressed by btc.

btc.a The downbranch ambiguity for node 1 (root node ADA, this is the total tree ambiguity).

btc.index The unique index of the BTC (used when the BTC is a constituent tree for a BHC).

btc.n This is an array of node records, defined next.

n.index The index of the BTC node using standard top-to-bottom left-to-right numbering starting with index 1 (see Figure 1).

n.inSuperClass An array of class labels (integers) that are associated with the decision to enter the node.

n.leftSuperClass An array of class labels (integers) that correspond to taking the left path out of the node.

n.rightSuperClass An array of class labels (integers) that correspond to taking the right path out of the node. The union of leftSuperClass and rightSuperClass is inSuperClass for each node.

n.A The ambiguity for node n.

n.u The right-path K-component feature vector.

n.v The left-path K-component feature vector.

n.wavelet String containing the wavelet class (mother wavelet type, e.g., coiflet).

n.w Specifies the K wavelet transform locations from which to obtain the desired K-component feature vector for comparison with n.u and n.v.

n.minTree Index of the BTC (tree) with minimum ambiguity corresponding node.

n.ada The average downbranch ambiguity for the node.

n.leftNode Index of the node corresponding to the left path out of n.

n.rightNode Index of the node corresponding to the right path out of n.

n.parent Index of parent node for n (set to 0 for the root node).

bhc.numBTC The number of constituent BTCs in the BHC.

bhc.btc An array of numBTC BTCs.

bhc.C The number of classes in the hypertree (classification problem size).

1. Obtain the training data sets for the C-class problem of interest and specify the allowable set of wavelets.
2. Specify the input data type, generating btc.cidType.
3. Construct a binary tree with $\log_2(C) + 1$ levels.
4. For each decision node, select left and right superclasses. Nominally, these are of equal size and consist simply of the leftmost elements for the left superclass and the rightmost elements for the right superclass.
5. Find the LDB for each wavelet type and decision node in the tree.
6. Select the most discriminating K elements of the most discriminating LDB for each node, generating btc.n.wavelet and btc.n.w for each decision node.
7. Find the average value of the K LDB elements for each superclass in each decision node, generating btc.n.u and btc.n.v for each decision node.
8. For each decision node n, compute the ambiguity, generating btc.n.A.
9. For each decision node n, compute the average downbranch ambiguity btc.n.ada.

Figure 5: The algorithm for constructing a BTC.

6.2 Classifier Construction Algorithms

In this section we present the basic construction algorithms. Because our fundamental feature-finding tool is an adapted LDB algorithm (see Appendix B), the classifiers must be constructed with the use of sufficient training data.

Figure 5 provides a simple statement of the algorithm for creating a basic BTC, and Figure 6 provides the algorithm for the BHC.

6.3 Classifier Tree Traversal Algorithms

In this section, we provide algorithm statements for traversing the BTC and BHC trees. That is, we discuss how to use the constructed trees to classify an input. For the BTC, the traversal is straightforward. For the BHC, there are variants that could exhibit substantially different performances.

1. Specify the set of available CIDs. This generates bhc.numBTC.
2. Obtain the training data sets for the C-class problem of interest and specify the allowable set of wavelets. This generates bhc.C.
3. For each distinct CID type, construct a BTC as in Figure 5. This generates the array bhc.btc.
4. Set n = 1.
5. For decision node index n, find the BTC indices for all corresponding nodes in the array of BTCs bhc.btc. Call this set of BTC indices T_n.
6. Find the BTC in T_n with minimum btc.n.A. Denote the index of this constituent BTC m.
7. Set bhc.btc.n.minTree equal to m for each BTC in the set T_n.
8. Increment n by 1. If n is less than $2C$, goto Step 5.

Figure 6: The algorithm for constructing a BHC.

6.3.1 Binary Tree Classifiers

The algorithm for traversing a BTC is shown in Figure 7. The basic idea is to compute the feature vector at a decision node and compare it to the left- and right-path stored average feature vectors for that node. If the measured feature vector more closely resembles the left (right) feature vector, then the left (right) path out of the node is taken. If the measured vector is equally well correlated with both the left and right vectors, then a fair two-sided coin is flipped.

6.3.2 Binary Hypertree Classifiers

The basic algorithm for traversing a BHC is provided in Figure 8.

7 Illustrative Example

Here we extend our basic eight-class toy problem originally defined in the DARPA TRUMPETS work [2]. The basic idea of the problem is, as before, a sort of image classification, but now we have multiple image modes available. So the classification problem will be to identify the label of the object that gives rise to the available image-oriented CIDs. Let's suppose we have a crude imager that produces binary-valued pixels (black-and-white camera), a slightly more sophisticated imager that produces gray-scale images (gray-scale camera), one that produces color images (color

1. Obtain the data to be classified (a CID $\boldsymbol{X}$).
2. Compute the classification tree depth $D = \log_2(C)$.
3. Set $n = 1$ and $j = 1$.
4. Set $c_m = \mathbf{W}_{j_{n,m},k_{n,m},l_{n,m}}[\boldsymbol{X}]$ for $m = 1, \ldots, K$, $\mathbf{c} = [c_1, \ldots, c_K]$.
5. Obtain the children of node n: $[n_l, n_r] = \text{children}(n)$.
6. Set $\rho_l = \text{CorrCoef}(\mathbf{c}, \mathbf{v}_n)$.
7. Set $\rho_r = \text{CorrCoef}(\mathbf{c}, \mathbf{u}_n)$.
8. if $\rho_l > \rho_r$ then $n = n_l$ else $n = n_r$.
9. $j = j + 1$.
10. If $j < D$ goto Step 4.
11. Class decision is $n - (2^D - 1)$.

Figure 7: The algorithm for traversing the BTC.

1. Obtain at least one CID $\boldsymbol{X}$ to be classified.
2. Choose the constituent BTC with minimum node-1 ambiguity, say the BTC with index j. Node 1 is the current node.
3. If no CID is present for the current BTC, request the CID from the sensor suite.
4. Use the basic BTC algorithm to select the left or right path out of current node in current BTC j, landing on node k. Update current node to k.
5. Switch to BTC bhc.btc[j].n[k].minTree, the minimum-ambiguity BTC for node k. Update current BTC to bhc.btc[j].n[k].minTree.
6. If node k is a decision node, goto Step 3.
7. Class decision is $k - (2^D - 1)$.

Figure 8: The basic algorithm for traversing the BHC.

1. Obtain at least one CID $\boldsymbol{X}$ to be classified.
2. Choose the constituent BTC with minimum node-1 ambiguity, say the BTC with index j. Node 1 is the current node. Set $j_0 = j$.
3. If no CID is present for the current BTC, request the K-component feature vector for the CID and current node from the sensor suite.
4. Use the basic BTC algorithm to select the left or right path out of current node in current BTC j, landing on node k. Update current node to k. Update current BTC j to j_0.
5. Switch to BTC bhc.btc[j].n[k].minTree, the minimum-ambiguity BTC for node k. Update current BTC to bhc.btc[j].n[k].minTree.
6. If node k is a decision node, goto Step 3.
7. Class decision is $k - (2^D - 1)$.

Figure 9: The switch-and-return algorithm for traversing the BHC.

camera), and one that produces infrared images (infrared camera), which images temperature differences in the underlying physical objects.

We define the classes in terms of the idealized images they produce at the outputs of the four cameras, as shown in Figure 10. The interesting part of this problem is to create a situation in which classification is ambiguous (flawed; irreducible nonzero probability of error) for each imager independently, but not when used together in a BHC (as needed during classification) or in a BSC.

Discussion.

A casual examination of the idealized images in Figure 10 reveals that there are several completely ambiguous classes for each camera type. For example, for the black-and-white camera, there are two sets of ambiguous classes: the circles and the crosses. Out of eight classes, only two may be reliably classified correctly. It is very important to realize that *these problems are inherently difficult for any classifier structure, not just tree-based structures*.

In addition to the large number of ambiguous classes for each camera, note that no two cameras possess the same set of ambiguous classes. That is, for any two classes, there is at least one camera type for which the corresponding images are distinct. This fact is crucial to the success of a BHC (or a BSC): there must be sufficient discriminatory power in the *collection* of CID types to allow good classification performance. If there is not, then the collection must be modified in some fashion, such as adding new CID types (sensor modalities or types).

Sample Hypertrees.

The best BTC for each camera type is shown in Figure 11. For this problem, balanced binary trees are not optimal for the gray-scale, color, and infrared camera types. For each of the four best

binary trees, additional trees are constructed using the obtained superclasses for each of the other three camera types. This ensures that each node in each of the four best trees has at least three corresponding nodes. The four sets of four BTCs are shown in Figures 12–15 for the black-and-white, gray-scale, color, and infrared cameras, respectively.

Note that no single constituent BTC is free from highly ambiguous nodes. However, through the use of the hypertree, we can find an unambiguous path through the collection of BTCs for each input class, guaranteeing good performance. The hypertree indices are not shown in the figures to keep the figures legible. Let us illustrate the BHC operation with a few examples next.

Operation.

First suppose that the true class label is 1, and we are provided initially with an output from the black-and-white (B&W) camera. So we know that we start in BTC 1 (see Figure 12). Node 1 of tree 1 points to itself, so the processing is performed on the B&W camera output, resulting in taking the left branch out of node 1 and landing on node 2 in tree 1. This node is ambiguous and points to tree 3, node 2. To continue, the ISP system must request a color camera output. It then processes this new sensor output and takes the left branch out of node 2, landing on node 4, which is a terminal node and is correct. The traversal of the hypertree can be summarized in tabular form in the following way:

(Tree, Node)	Camera	Get New Data?	Branch To	Jump To
(1, 1)	B&W	No	(1, 2)	
(1, 2)	B&W	No		(3, 2)
(3, 2)	Color	Yes	(3, 4)	

Next consider that the true class label is 2 and we again start out with a B&W camera output. The following hypertree traversal results.

(Tree, Node)	Camera	Get New Data?	Branch To	Jump To
(1, 1)	B&W	No	(1, 2)	
(1, 2)	B&W	No		(3, 2)
(3, 2)	Color	Yes	(3, 5)	
(3, 5)	Color	No		(12, 5)
(12, 5)	Gray-Scale	Yes	(12, 9)	
(12, 9)	Gray-Scale	No		(13, 9)
(13, 9)	IR	Yes	(13, 14)	

We see that to properly classify class 2 starting with a B&W image, we require all three additional camera outputs. The situation is quite different for classes 3 and 4. For example, for class 3 and an initial image from the B&W camera, we have the following hypertree traversal:

(Tree, Node)	Camera	Get New Data?	Branch To	Jump To
(1, 1)	B&W	No	(1, 3)	
(1, 3)	B&W	No	(1, 6)	
(1, 6)	B&W	No	(1, 12)	

For this example, no additional camera outputs are required for good classification. A similar result holds for class 4.

Notice that if we have a class-3 input and we begin with an IR-camera output, we do, in fact, require additional camera outputs to correctly classify the input:

(Tree, Node)	Camera	Get New Data?	Branch To	Jump To
(4, 1)	IR	No	(4, 2)	
(4, 2)	IR	No	(4, 4)	
(4, 4)	IR	No		(16, 4)
(16, 4)	Gray-Scale	Yes	(16, 8)	

8 Extensions

The tree-based classification methods outlined in this report are quite general in two important respects. First, they can accommodate widely differing input (sensor output) data types. For example, the available sensor outputs can consist of two-dimensional optical images, two-dimensional range-doppler radar returns, one-dimensional high-range resolution (HRR) returns, SAR images, infrared images, sound records, etc. In other words, the hypertree classification architecture is naturally suited for problems in which *data fusion* is essential.

The second way in which the classifiers exhibit great generality and flexibility is in the specific choice of classifier type. We need not restrict ourselves to tree-based classifiers in order to reap the benefits of the ISP-enabled hypertree system. All that is required is the ability to detect ambiguous decisions and a way to link ambiguous decisions to the collection of the best new data set for resolving the detected ambiguity.

9 Conclusions

We have documented our initial research efforts in the area of binary hypertree classifiers (BHCs). The core notion is the generalization of simple binary-tree classifiers (BTCs) to ISP systems in which traversing the tree is linked to sensor controls so that as ambiguous situations are encountered, further sensor data is requested from the sensor with the most discriminatory power for the situation. In this way processing (feature-based classification) is strongly integrated with sensing.

A mathematical framework for hypertree classifiers is laid out, some performance analysis results are obtained, and algorithms for tree construction and traversal are presented. Future work will focus on creating a simulator and extending the mathematical analysis.

References

[1] “A Mathematical Methodology for Managing and Integrating Sensors and Processors in Distributed Systems for Radar and Communications,” Mission Research Corporation Proposal for the DARPA ISP Program, October 2001.

[2] C. M. Spooner and G. K. Yeung, “Local Discriminant Bases for TRUMPETS ATR,” DARPA TRUMPETS Program, MRC Technical Report, November 2000.

[3] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, Wiley & Sons, New York, 2001.

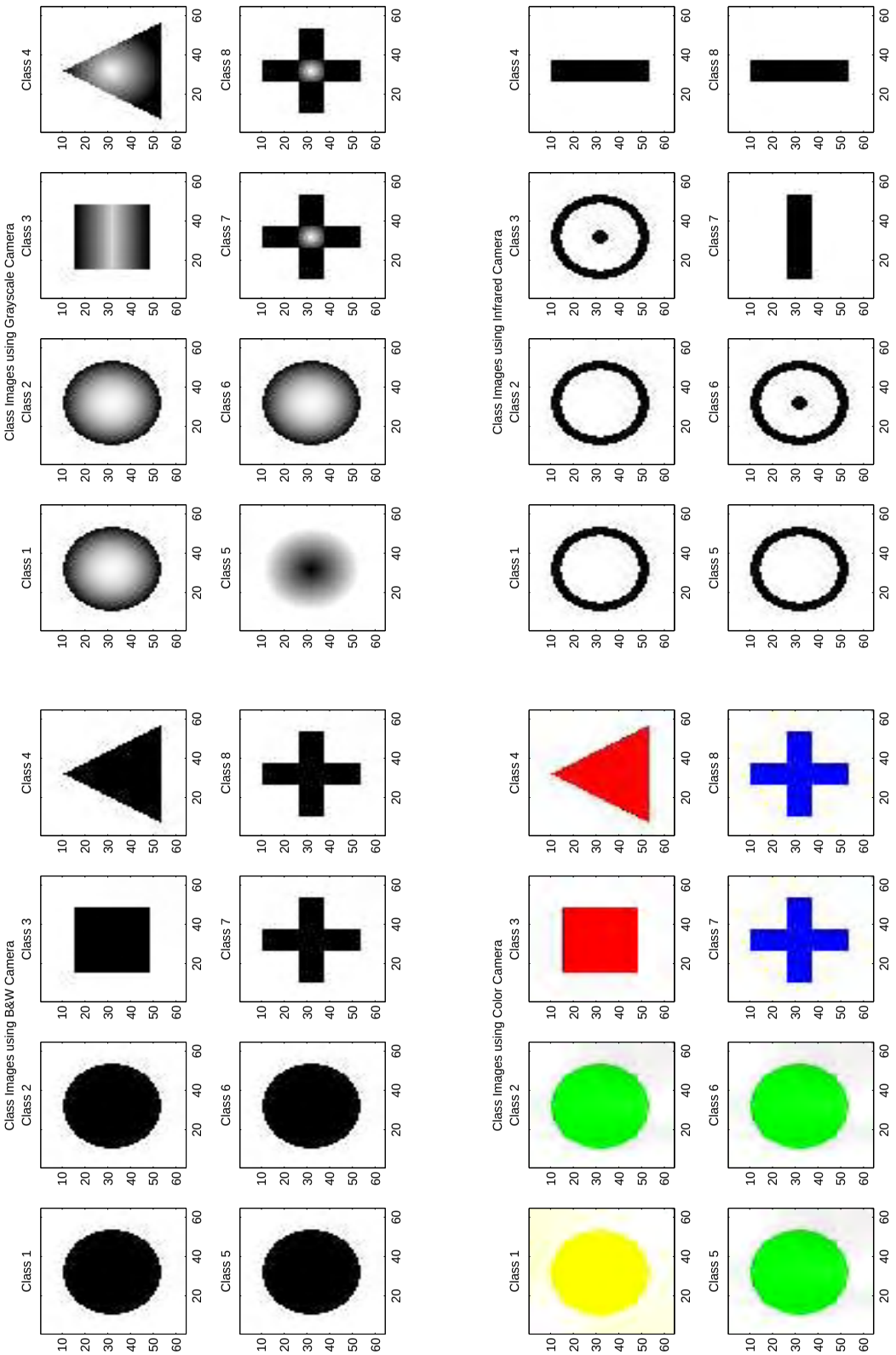


Figure 10: Idealized images for the eight-class illustrative example.

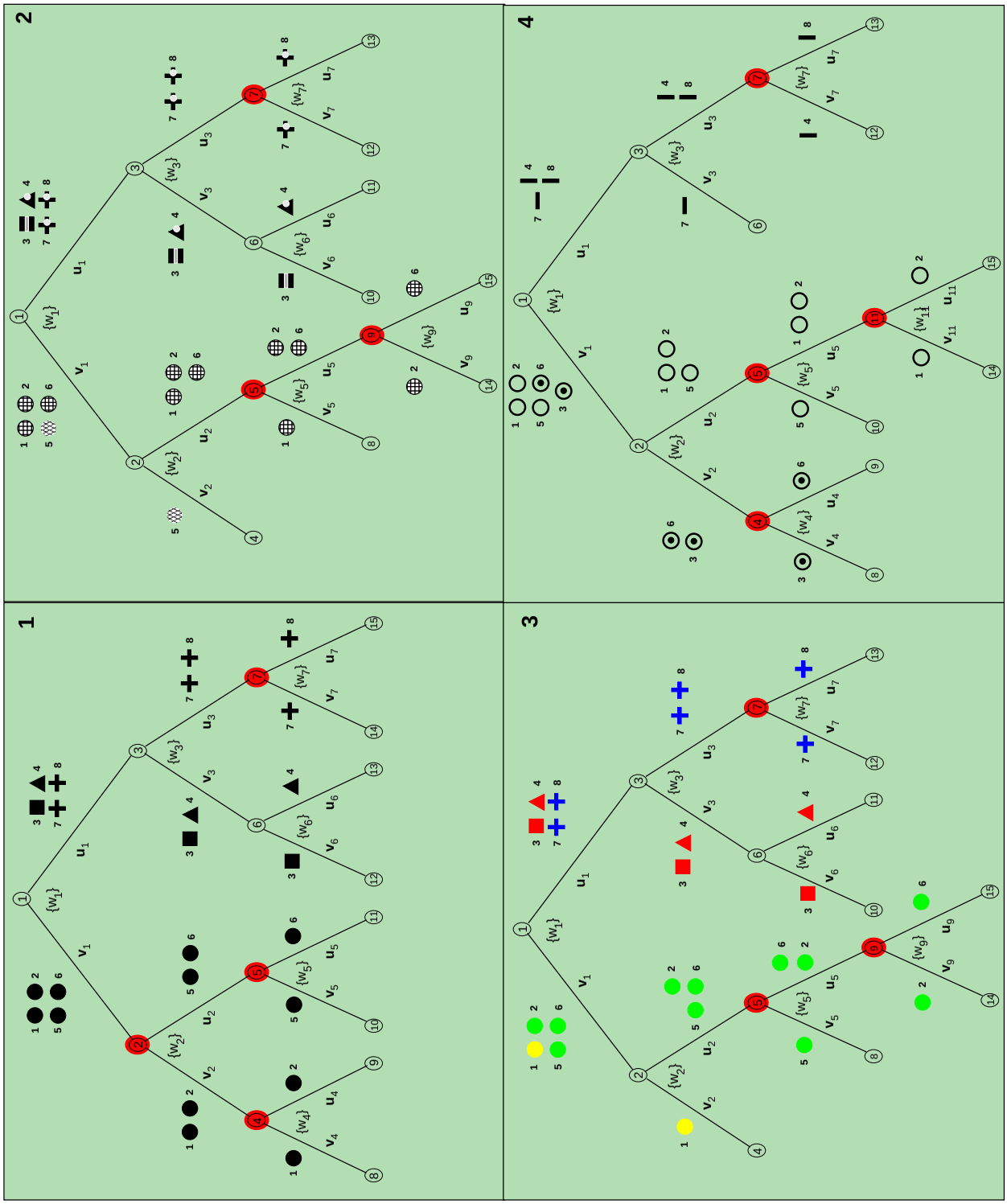


Figure 11: The best binary-tree classifiers for each of the four camera types. The ambiguous nodes are highlighted in red.

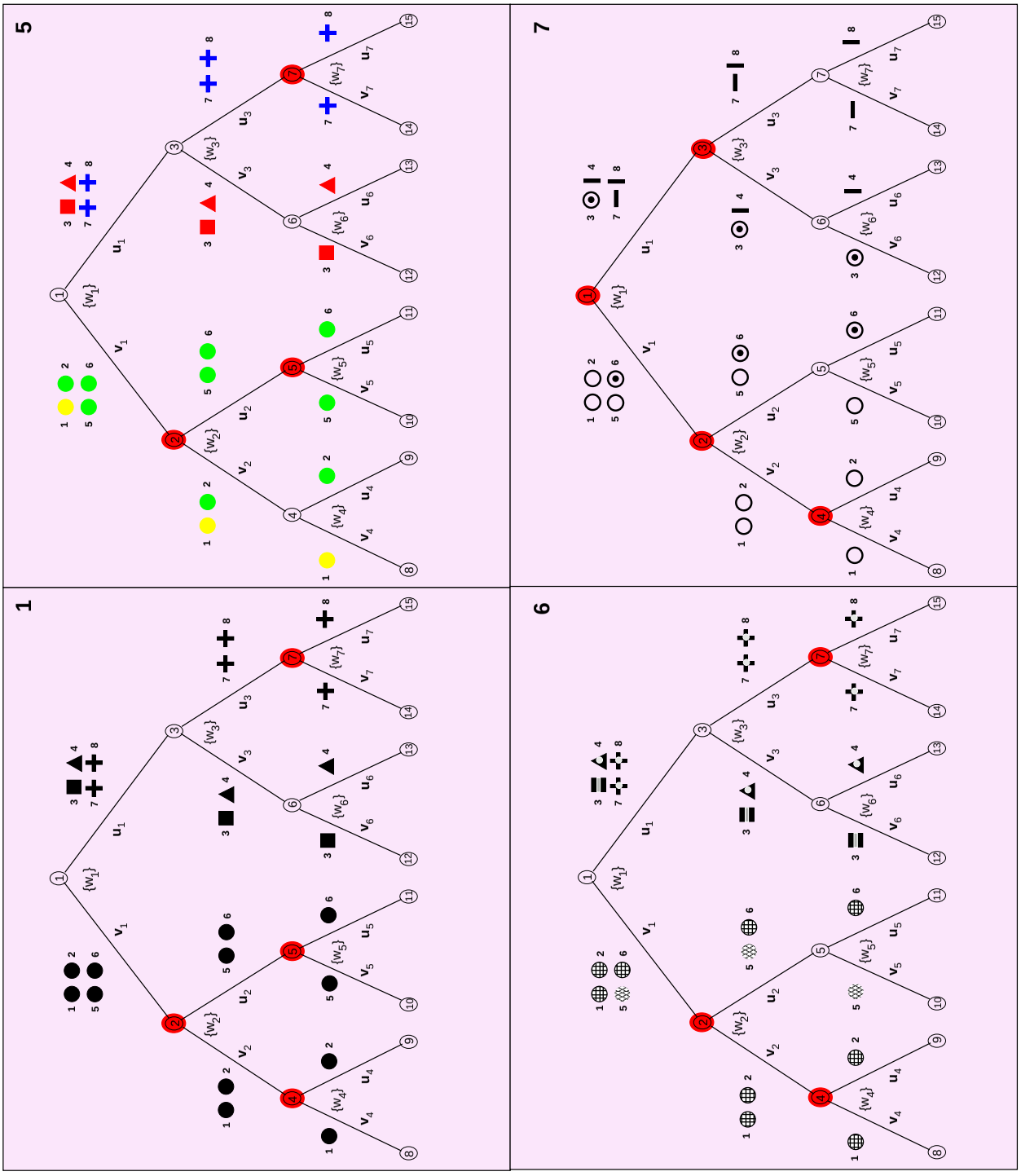


Figure 12: The binary-tree classifiers for the tree specified by the best BTC for the black-and-white camera. Ambiguous nodes are highlighted in red.

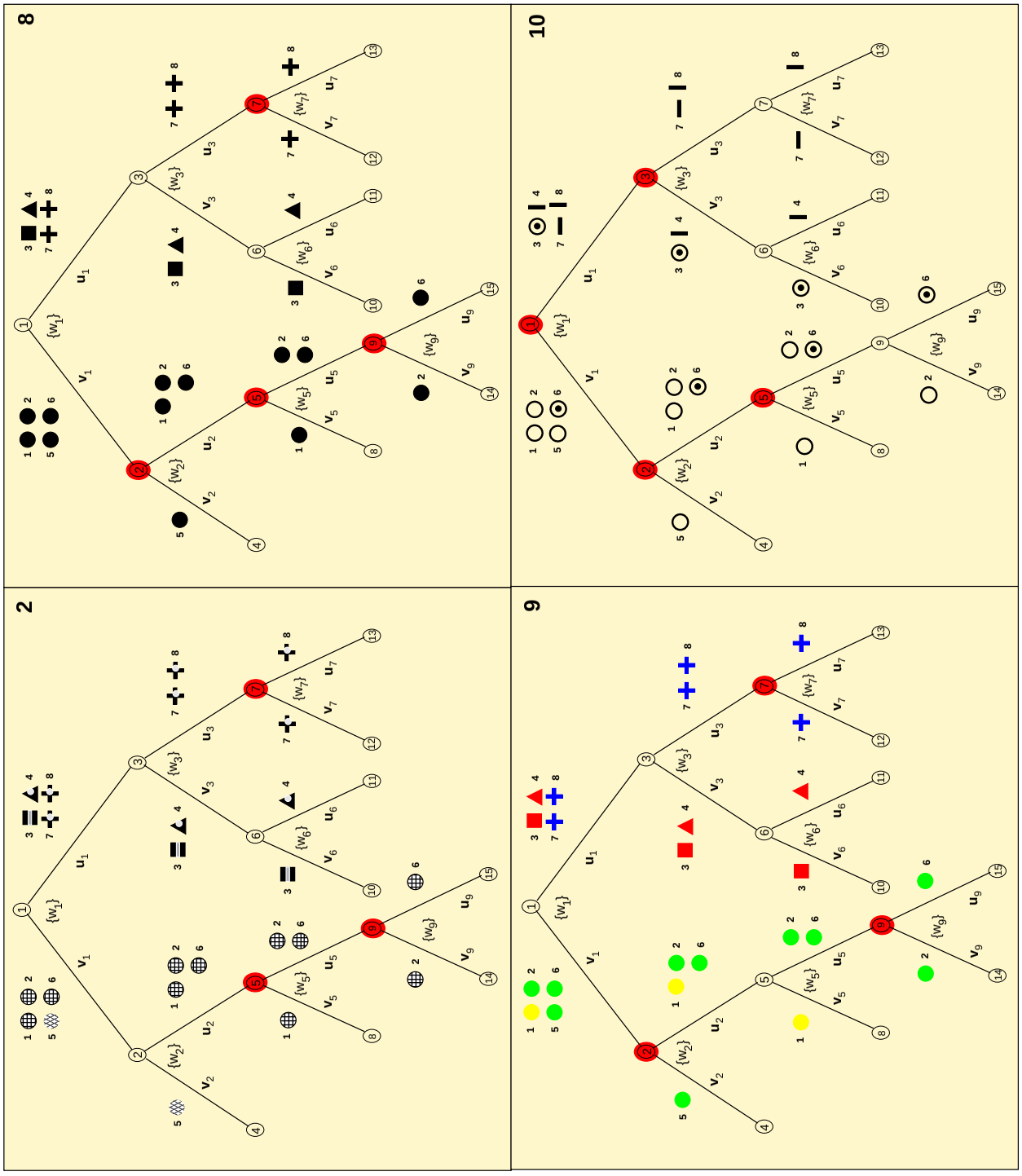


Figure 13: The binary-tree classifiers for the tree specified by the best BTC for the gray-scale camera. Ambiguous nodes are highlighted in red.

147

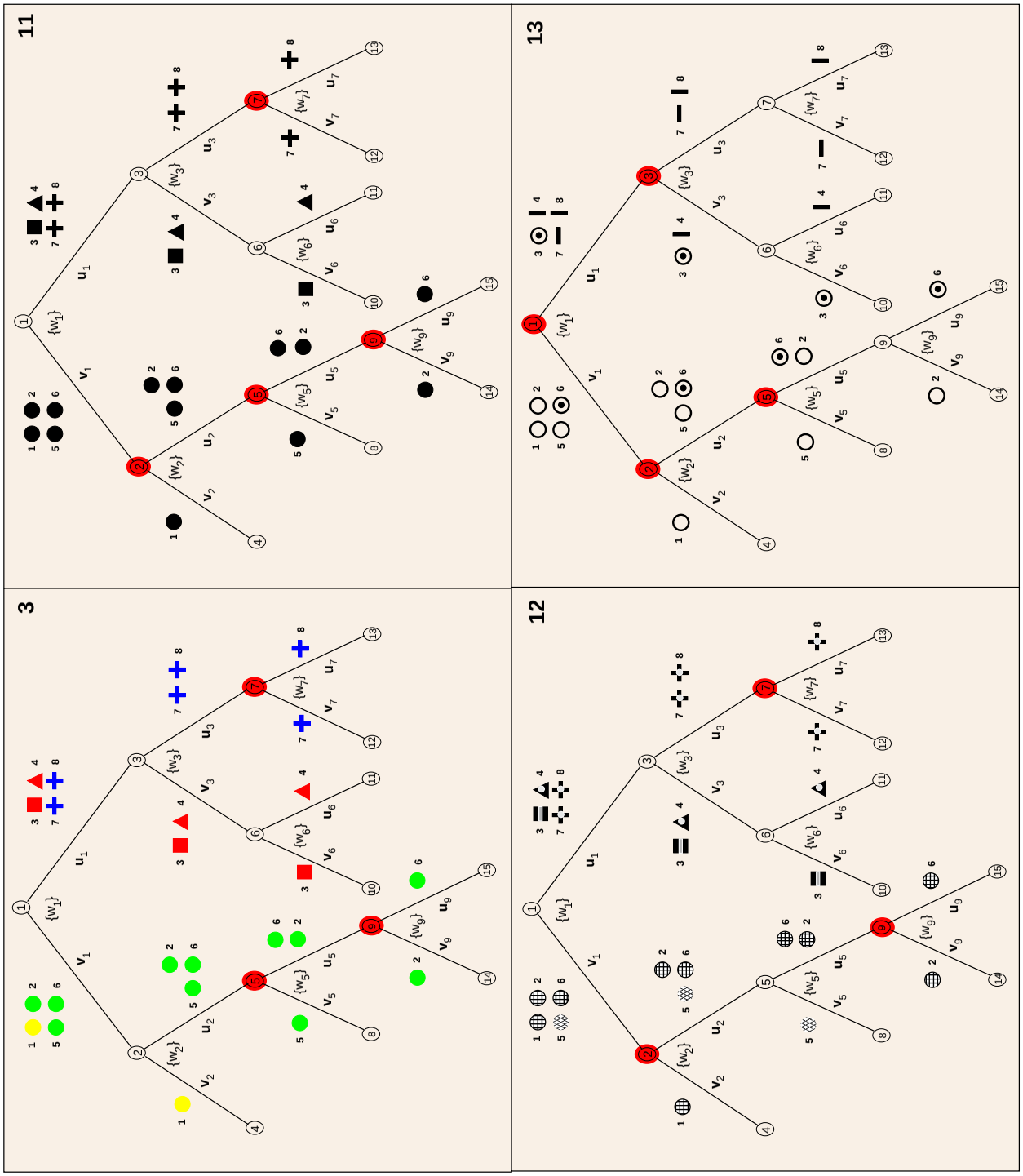


Figure 14: The binary-tree classifiers for the tree specified by the best BTC for the color camera. Ambiguous nodes are highlighted in red.

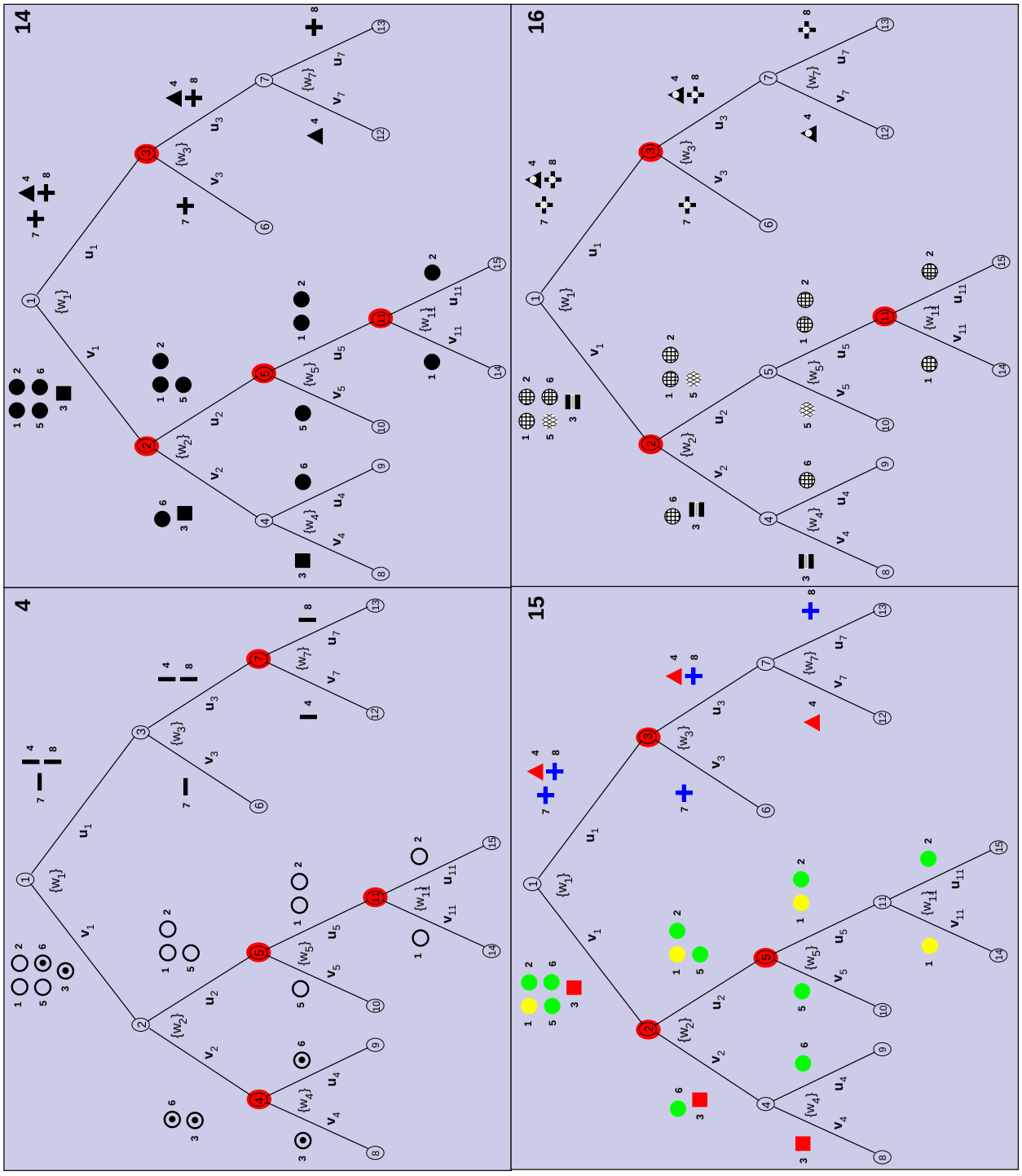


Figure 15: The binary-tree classifiers for the tree specified by the best BTC for the infrared camera. Ambiguous nodes are highlighted in red.

149

[4] N. Saito, “Local Feature Extraction and its Applications using a Library of Bases,” Ph.D. Dissertation, Yale University, December 1994.

[5] S. Mallat, *A Wavelet Tour of Signal Processing,* Academic Press, San Diego, CA, 1998.

[6] `http://www-stat.stanford.edu/~wavelab.`

Appendices

A Wavelets and Wavelet Packets

Wavelet Transforms.

The *wavelet transform* of an $N \times N$ image is defined by a pair of quadrature mirror filters (QMFs) $h(\cdot)$ and $g(\cdot)$ and a maximum decomposition depth J [5]. The filters h and g are low- and high-pass filters, respectively. The transform iteratively applies the filters to the rows and columns of the image, subsamples the results, and then starts over with the subsampled data. The filters are applied in all four of their row-column combinations: low-low (LL), low-high (LH), high-low (HL), and high-high (HH). At each iteration, the convolution-sampling operation is applied to the LL data only, while the other three data sets are retained as is. At the final stage (stage J) the LL coefficients are also retained. For example, Figure 16 shows an image and its decomposition for $J = 1$.

It turns out that this iterative filtering and decimating process corresponds to data decomposition using a set of images that form a basis for all square-summable images. Let these *basis images* be denoted by $\mathbf{v}(j,k,l)$, $j = 0,1,\ldots,J$, $k = 0,1,\ldots,k(j)$, and $l = 0,1,\ldots,4^{n_0-j}-1$, where $N = 2^{n_0}$ is a dyadic number. Then the image data, denoted by $\mathbf{x}$, can be represented by

$$\mathbf{x} = \sum_{j,k,l} c_{j,k,l}\mathbf{v}(j,k,l), \tag{2}$$

where $c_{(\cdot)}$ is a set of coefficients. The variable index maximum $k(j)$ is equal to 0 for $j = 0$, to 3 for $j = J$, and to 2 otherwise.

Throughout the remainder of this report, the terms *basis image* and *basis vector* are used interchangeably. This usage emphasizes the strong connections to vector-space ideas and underscores that most of the discussion is applicable to D-dimensional data sets for $D > 2$.

Wavelet Packets.

The *wavelet packet decomposition* of an image is closely related to the wavelet transform. Instead of iteratively applying the QMFs to the LL data only, they are iteratively applied to each of the four data sets LL, LH, HL, and HH. This results in a set of vectors that contains many distinct bases, including the basis corresponding to the wavelet transform. Let this large set of linearly dependent vectors be denoted by $\mathbf{w}(j,k,l)$, $j = 0,1,\ldots,J$, $k = 0,1,\ldots,4^j-1$, $l = 0,1,\ldots,4^{n_0-j}-1$, and let the operator $\mathbf{W}_{j,k,l}$ denote the transformation of the image data $\mathbf{x}$ to the coefficient that corresponds to the basis image $\mathbf{w}(j,k,l)$,

$$c_{j,k,l} = \mathbf{W}_{j,k,l}[\mathbf{x}].$$

Then the image data is represented by

$$\begin{aligned}\mathbf{x} &= \sum_{j,k,l} c_{j,k,l}\mathbf{w}(j,k,l) \\ &= \sum_{j,k,l} \left(\mathbf{W}_{j,k,l}[\mathbf{x}]\right)\mathbf{w}(j,k,l).\end{aligned}$$

The image subspace spanned by the vectors $\mathbf{w}(j,k,\cdot)$ is denoted by $B(j,k)$. Each node in the tree of Figure 17 is associated with a single subspace $B(j,k)$. An orthogonal basis is an orthogonal collection of the $B(j,k)$ that spans the image space.

The wavelet packet decomposition is illustrated in Figure 17. Note that the wavelet transform consists of all extreme-left nodes and their immediate children.

An Alternate Subspace-Indexing Scheme.

In the previously described indexing scheme for $\mathbf{w}(j,k,l)$, the variable j denotes the depth in the decomposition tree (Figure 17), k denotes the node number at depth j (starting with $k=0$ on the left), and l denotes the particular element in the matrix associated with node k at depth j. For example, the filled node in Figure 17 corresponds to $j=2$, $k=14$, and it is associated with an 2^{n_0-2} by 2^{n_0-2} matrix, whose elements are indexed by l (starting with $l=0$) after creating a vector by concatenating its rows.

In computer programs that implement decomposition trees like that in Figure 17 (see WaveLab [6]), it can be more convenient to use a different indexing scheme. In this alternate scheme, the basis images are also indexed by an ordered triplet (j,k,l), where j denotes the absolute node number, and the pair (k,l) denotes the matrix element associated with node j. For example, the filled node in Figure 17 corresponds to $j=20$, and it is associated with an 2^{n_0-2} by 2^{n_0-2} matrix, so that $k=1,\ldots,2^{n_0-2}$, and $l=1,\ldots,2^{n_0-2}$.

We refer to the first numbering system as Saito's numbering system (SNS) [4] and to the alternate as the wavetree numbering system (WNS), after MRC's MATLAB data structure `wavetree`.

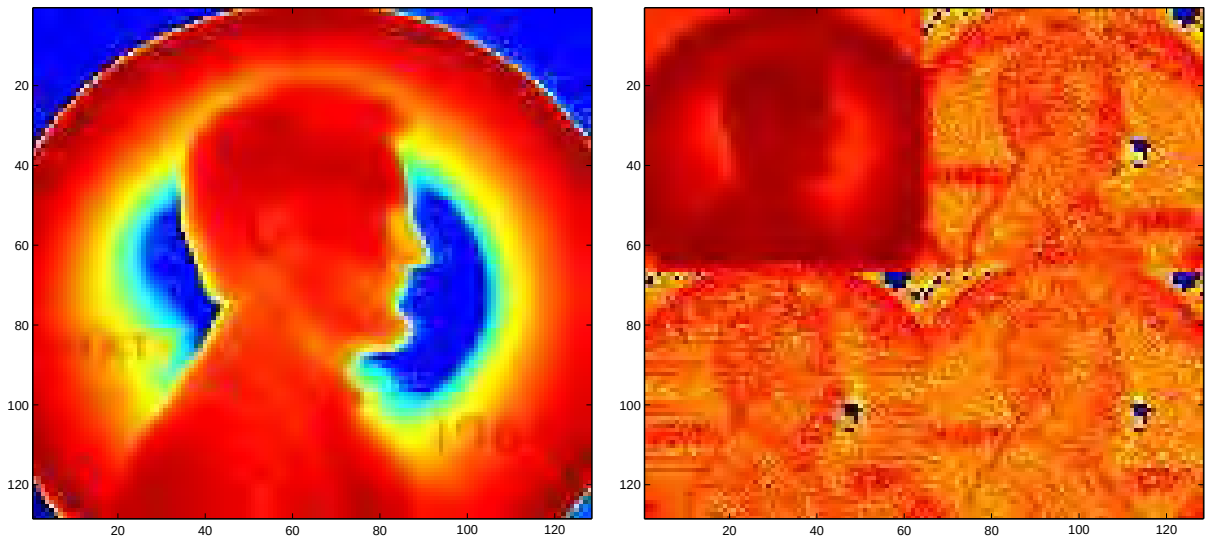


Figure 16: Wavelet transform for $J=1$, which is also the wavelet packet for $J=1$. The Daubechies wavelet with parameter 20 is used.

Analogy to the Two-Dimensional Fourier Transform.

The representation of the image $\mathbf{x}$ as a weighted sum of basis images might be better understood by analogy with the simpler and more familiar two-dimensional discrete Fourier transform (2D-FT).

The 2D-FT for the image $\mathbf{x} = \{x(u,v)\}$ is

$$y(f_x, f_y) = \sum_{u=1}^{N} \sum_{v=1}^{N} x(u,v) e^{-i2\pi f_x u/N} e^{-i2\pi f_y v/N},$$

and its inverse is simply

$$x(u,v) = \sum_{f_x=1}^{N} \sum_{f_y=1}^{N} y(f_x, f_y) e^{i2\pi f_x u/N} e^{i2\pi f_y v/N}.$$

A single element from the inverse-transform sum is

$$y(r,s) \left[e^{i2\pi r u/N} e^{i2\pi s v/N} \right],$$

which is a scaled image in u and v. The image is trivially obtained by inverse transforming the 2D-FT that is zero everywhere except for $f_x = r, f_y = s$.

The wavelet transform, or any other transform based on the wavelet packet, works in a similar way, although the transform is more complex. However, we can obtain the basis images by inverse transforming a set of coefficients that are all zero except in the desired location (j,k,l).

Wavelet-Packet Processing.

The goal in wavelet-packet compression is to search over all possible bases corresponding to a wavelet packet for the one that possesses maximum *energy compaction* with respect to a target class of interest. This simply means that the energy of the decomposed data is concentrated in the fewest possible basis coefficients. If the number of significant basis coefficients is small compared to N^2, then by representing the data as the values of these few coefficients, a large degree of compression is obtained at a small loss in fidelity.

The goal in wavelet-packet classification is to search over all possible bases for the one that possesses the basis vectors that have the maximum *discrimination power* over all input target classes of interest. The discrimination power is quantified by a distance measure. Such a basis is referred to as a *local discriminant basis* (LDB) [4]. The ideal LDB is one for which a very small number (compared to N^2) of basis coefficients can be used to reliably determine the class to which an input data set belongs. LDBs, and how to obtain them, are the focus of the next section.

B Local Discriminant Bases

The material in this appendix is excerpted from [2].

LDBs are obtained only in the context of a specific set of target classes of interest. Suppose we have C classes of interest, and we have N_c training images for class c, $c = 1, \ldots, C$, provided in sets X_c. It is assumed that the training images are representative of their respective classes. The fundamental idea behind the LDB is to find a basis such that there are a few basis vectors whose coefficients vary widely among the classes while varying little between members of a class. Before we can determine whether a vector can provide good discrimination between classes, we need to know something about the behavior of the vector's coefficients within each class. Specifically, we

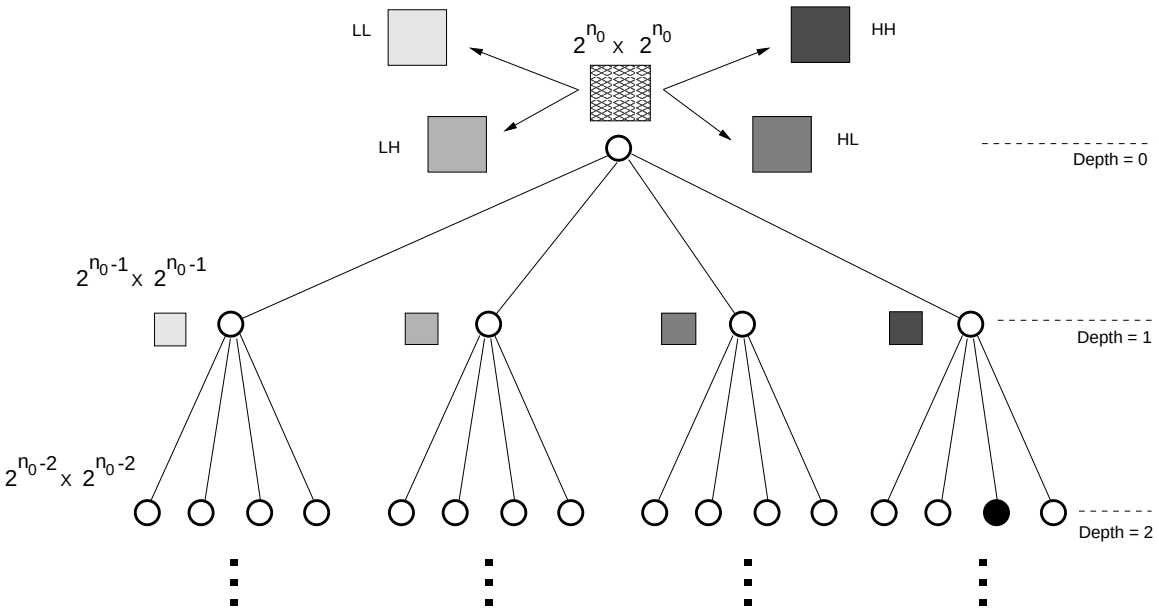


Figure 17: Illustration of the wavelet packet decomposition.

need to establish a measure of the average strength of the coefficients throughout the packet for each class. The strength measure could be average value, average absolute value, average energy, and others [4]. Then we need to establish a measure of the distance or difference between the basis coefficients for two or more classes; this distance measures the discrimination power of the corresponding basis image.

Coarse LDB Algorithm.

To find the LDB, and the best vectors in the LDB, we perform the steps shown in Figure 18, which we shall expand upon in the remainder of this section.

1. Obtain the training data sets.
2. Select a wavelet by choosing a QMF pair.
3. For each target class, find the energy in its average wavelet packet decomposition using the training data.
4. Search over the bases in the packet for the one with the largest differences in average interclass energy.
5. For the selected basis, order the basis vectors by their discrimination power.

Figure 18: A coarse statement of the algorithm for finding the local discriminant basis (LDB).

B.1 Constructing the Average Packet-Energy Map

The goal of this step in the LDB algorithm is to characterize the average energy contained in the subspaces $B(j,k)$ for each target class. This gives us an idea of where the energy is concentrated for the various classes, which can be used to select subspaces that possess greatly varying average energies over the input classes, indicating good discrimination power.

The baseline average packet-energy map [4] for class c is defined by $\Gamma_c(j,k,l)$:

$$\Gamma_c(j,k,l) \triangleq \frac{1}{E_c} \sum_{i:\mathbf{x}_i \in X_c} \left(\mathbf{W}_{j,k,l}[\mathbf{x}_i]\right)^2, \tag{3}$$

where the *class energy* E_c is defined by

$$E_c \triangleq \sum_{i:\mathbf{x}_i \in X_c} \|\mathbf{x}_i\|^2.$$

The energy map Γ_c has the advantages of conceptual and mathematical simplicity, and it does indicate the subspaces that are particularly energetic, but it has drawbacks for LDB-based ATR. In particular, it cannot distinguish between subspaces that have similar energy by coefficients with opposing signs; such pairs of subspaces may in fact be very useful for classification. To avoid this drawback, an alternate energy measure is simply the *average value map*:

$$\alpha_c(j,k,l) \triangleq \frac{1}{N_c} \sum_{i:\mathbf{x}_i \in X_c} \mathbf{W}_{j,k,l}[\mathbf{x}_i]. \tag{4}$$

This map can be especially useful in conjunction with the *variance map*:

$$V_c(j,k,l) \triangleq \left[\frac{1}{N_c} \sum_{i:\mathbf{x}_i \in X_c} \left(\mathbf{W}_{j,k,l}[\mathbf{x}_i]\right)^2 - \alpha_c(j,k,l)^2\right]^{1/2}. \tag{5}$$

Because there are competing energy-measuring functions, the generic energy-map function is denoted by $E_c(j,k,l)$. The energy for a subspace $B(j,k)$ is simply the sum of energies of its components.

B.2 Searching for the Best Discriminant Basis

The goal of this processing step is to use the energy maps to determine the LDB. The idea, as previously stated, is to find the nodes in the packet tree (or, equivalently, the subspaces $B(j,k)$) that possess a distinctly different average energy for each class. To find these nodes, we need to define a measure of distance between the subspace energy functions $\{E_c(j,k,\cdot)\}_{c=1}^{C}$. To be general, we base the distance measure $D(\cdot)$ on a pairwise distance measure $D_p(\cdot,\cdot)$:

$$D(\{y_i\}_{i=1}^{M}) = \sum_{j=1}^{M} \sum_{k=j+1}^{M} D_p(y_j, y_k) \tag{6}$$

The pairwise distance measure can be Euclidean distance, relative entropy, or others (see [4], page 66).

The baseline subspace energy distance function [4] can now be defined:

$$D(\{E_c(j,k,\cdot)\}_{c=1}^{C}) = D(\{E_c(j,k)\}_{c=1}^{C}) = \sum_{l=0}^{4^{n_0-j}-1} D(\{E_1(j,k,l), E_2(j,k,l), \ldots, E_C(j,k,l)\}).$$

This is the sum over all unique pairs of the pairwise distances for elements of the subspace $B(j,k)$. Note that if the energy in the vectors for each class is distinct from the energy in the other classes, the sum of pairwise differences will be large.

Denote the LDB by the collection of subspaces $A(j,k)$ and denote the children of subspace $B(j,k)$ by $\{B(j+1,i) : i \in I(j,k)\}$. The algorithm for obtaining the LDB is stated in Figure 19. Step 6 is the key step. The idea is to compare the discrimination power of a subspace to the sum of the discrimination powers of all its children. If the children are better discriminators, then they determine the composition of the best-basis subspace, but if the subspace itself is a better discriminator, then it is retained as the best-basis subspace.

1. Choose the desired QMFs for the wavelet packet of interest.
2. Specify the maximum decomposition depth J.
3. Specify the pairwise distance measure $D_p(\cdot,\cdot)$ and the subspace energy measure $E_c(j,k,l)$.
4. Compute the energy maps E_c for $c = 1, 2, \ldots, C$.
5. Set $A(J,k) = B(J,k)$ and $\Delta(J,k) = D(\{E_c(J,k)\}_{c=1}^{C})$ for $k = 0, 1, \ldots, 4^J - 1$. This initializes the LDB to the set of subspaces at the lowest level of the decomposition (viewed in terms of a tree).
6. Find the best subspace $A(j,k)$ for $j = J-1, \ldots, 0$ and $k = 0, 1, \ldots, 4^j - 1$ by using the following rule:

   Set $\Delta(j,k) = D(\{E_c(j,k)\}_{c=1}^{C})$.

   If $\Delta(j,k) \geq \sum_{n \in I(j,k)} \Delta(j+1,n)$

   then set $A(j,k) = B(j,k)$

   else set $A(j,k) = \bigcup_{n \in I(j,k)} A(j+1,n)$ and set $\Delta(j,k) = \sum_{n \in I(j,k)} \Delta(j+1,n)$.
7. The LDB is $A(0,0)$.

Figure 19: A detailed statement of the algorithm used to find the local discriminant basis (LDB).

B.3 Ordering Basis Vectors by Discriminant Power

The final step in obtaining the LDB is to order the individual basis vectors in terms of their discrimination power. A natural choice for the measure of discrimination power is the distance measure used in obtaining the best basis. When obtaining the best basis, the measure was applied to subspaces $B(j,k)$. Here, the measure is applied to the individual vectors in the subspaces making up the best basis. The vectors are then sorted in decreasing order with respect to their distance.

An alternative to using the best-basis distance measure to order the basis vectors involves using the average-value energy map $\alpha_c(j,k,l)$ together with the variance map $V_c(j,k,l)$. The primary difficulty with using the baseline energy map Γ_c together with the distance measure D_p is that a vector that possesses a high average energy for a target class can also have a high variance for the images *within* the class, which can substantially limit its applicability for classification. The ideal vector has the following properties

1. Maximum mean values over the C target classes,
2. Maximally distinct mean values over the C classes,
3. Small variance for each class.

Motivated by these considerations, we can define a distance between two elements of the best basis that is particularly suitable for classification purposes. For classes p and q, define

$$\begin{aligned} m_i &\triangleq \alpha_i(j,k,l), \\ s_i &\triangleq V_i(j,k,l), \end{aligned}$$

for $i = p, q$. Then the pairwise distance is defined by

$$D_v(\mathbf{w}(j,k,l),p,q) \triangleq \begin{cases} (m_q - s_q) - (m_p + s_p), & m_q > m_p \\ (m_p - s_p) - (m_q + s_q), & m_q \leq m_p. \end{cases}$$

The distance between more than two classes can be the sum of the pairwise distances,

$$D(\mathbf{w}(j,k,l)) = \sum_{p=1}^{M} \sum_{q=p+1}^{M} D_v(\mathbf{w}(j,k,l),p,q),$$

or the minimum over all pairs

$$D(\mathbf{w}(j,k,l)) = \min_{p,q} D_v(\mathbf{w}(j,k,l),p,q).$$

Binary Hypertree Classifiers for ATR

Experimental Study

Chad M. Spooner
Mission Research Corporation

March 8, 2005

Version 1.0

Abstract

Recently developed binary-tree-based classifiers are studied through simulation experiments. The two distinct classifiers are general and are applicable to a wide variety of classification problems. The classifiers accept one- and two-dimensional inputs and can be easily generalized to higher dimensions. The core classifier is the *binary tree classifier* (BTC). This classifier employs the *local discriminant basis* (LDB) to automatically and jointly determine the best tree topology and feature-vector values for each decision-node in the tree. The *binary hypertree classifier* (BHC) combines several BTCs for the special situation in which multiple input data types are available for each class. For example, the objects may be viewed with one of several distinct camera types. A BTC is created for each camera type and the BHC combines these in an efficient manner so that performance is maximized while using a minimum of input data types. The performance of these and related classifiers is evaluated in this report via simulation experiments for one- and two-dimensional inputs. It is shown that the classifiers are very good at automatically determining the structure of a given problem and extracting the most useful feature subsets for inclusion in the tree structures.

Contents

List of Figures

List of Tables

1 Introduction

The ability to automatically determine the type of a remote target, such as a tank, bus, or armored vehicle, is crucial to successful military operations because it helps sort out friends, foes, and neutrals in the chain of weapons-targeting operations. The subject of this report is a new class of systems for automatic target recognition (ATR) that is developed mathematically in a companion report [1]. The new systems employ the local discriminant basis (LDB) [3] and *binary- tree classifiers* in an attempt to automatically identify and characterize the highly discriminatory data elements and, simultaneously, minimize the required number of sensor modalities for accurate classification. In other words, the systems attempt to reap the performance gains of a full-blown sensor-fusion approach by selectively requesting new data or modalities only when required to eliminate a class ambiguity.

This report presents a set of experimental results aimed at proof-of-concept for the binary tree classifiers presented in [1]. We aim to answer the following questions. Can the developed structures automatically determine the low-dimensional data subspace that provides optimal classification performance? Can the developed hypertree structures minimize or substantially reduce the amount of data needed for a given performance level when compared to a classifier that fuses all available sensor data? To provide the proof-of-concept and answers to these questions, we apply the developed machinery to several problems involving simulated one- and two-dimensional data sets, as well as to publicly available data sets used in classification-system research and development.

The remainder of this report is organized as follows. The binary-tree classification structures are reviewed in Section 2, and the methods of training the classifiers are described in Section 3. Classifier operation (a relatively simple task compared to training) is briefly described in Section 4 (see also [1]). The various experiments and results are described in Section 5, and concluding remarks are provided in Section 6.

2 Binary Trees for Classification

In this section, we review the three tree-based classifier structures developed in [1]: the binary tree classifier (BTC), binary supertree classifier (BSC), and the binary hypertee classifier (BHC). The BTC is a classifier that uses a single sensor modality (called here a classifier input data (CID) type) and, usually, one wavelet type. Its topology and the classes involved at the decision nodes can be fixed in advance of training or chosen jointly during training. The BSC is a BTC that uses all available CIDs types (concatenated together) as its input; it represents a fusion algorithm. The BHC links together an arbitrary number of BTCs or BSCs to form a structure that jumps from CID to CID *as needed* or, more generally, from BTC to BTC as needed. The performance of the BHC is, in some cases, equal to that of the BSC, but the average data requirement should be smaller.

2.1 The Binary Tree Classifier (BTC)

The binary tree classifier has structure as illustrated by the example in Figure 1. The tree can be balanced or unbalanced; the particular connections of decision and terminal nodes is called the *tree topology*. For each decision node in the BTC, there are several quantities that need specification:

1. *Measurement Vector* $\mathbf{w}$. This vector has length K and specifies the K wavelet operations used to make the binary decision at the node.
2. *Left Superclass Set L and Right Superclass set R.* Each decision node makes a single binary decision between the *superclass*es on the left and right. The union of L and R for any decision node is equal to the set of all classes inherited from the node's parent. For example, refering to Figure 1, node 1 (the root node) splits the class-label set $\{1, 2, 3, 4, 5, 6, 7, 8\}$ into nonempty subsets $L_1 = \{1, 2, 3, 4, 5\}$ and $R_1 = \{6, 7, 8\}$.
3. *Left Average Feature Vector* $\mathbf{v}$. This vector has length K and is the average value of the K measurements specified by $\mathbf{w}$ using the classes contained in superclass set L.
4. *Right Average Feature Vector* $\mathbf{u}$. This vector has length K and is the average value of the K measurements specified by $\mathbf{w}$ using the classes contained in superclass set R.

There are many possible tree topologies, and for each of these, there are many choices for superclass selection. The idea is to jointly select the topology, superclasses, and measurement vectors such that the tree has optimal performance. The code developed during this work is general enough to accommodate a fixed topology, a fixed topology with fixed superclasses, or a free topology and free superclass selection.

2.2 The Binary Supertree Classifier (BSC)

The binary supertree classifier is simply a BTC with a multimodal classifier input data (CID) type. The data for each CID is concatenated to form the "super" input to the BSC.

2.3 The Binary Hypertree Classifier (BHC)

The binary hypertree classifier connects two or more BTCs that are aimed at solving the same classification problem. A graphical depiction of a BHC is shown in Figure 2. This particular BHC comprises four BTCs, each associated with a unique CID. The key idea is that each decision node in each BTC has an *ambiguity* associated with it such that for low ambiguity (near zero), the decision made at the node is almost always correct (low probability of error), and for high ambiguity (near 1), the decision is wrong with probability near one-half. Some BTCs may have several decision nodes with very low ambiguity and several with high ambiguity. The BHC points each node of each BTC to the constituent BTC having the node with lowest ambiguity and same union of L and R.

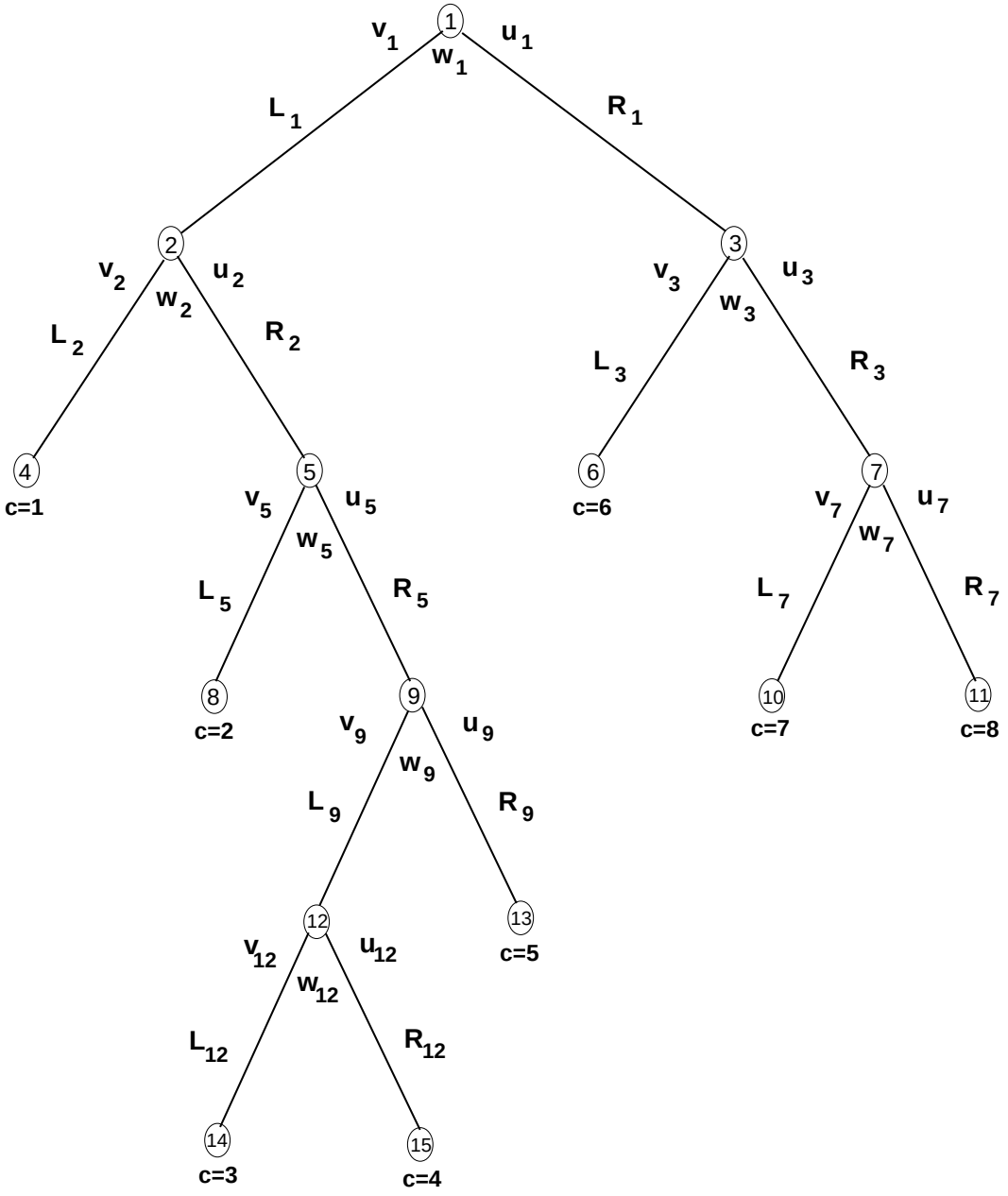


Figure 1: A typical BTC for an eight-class problem.

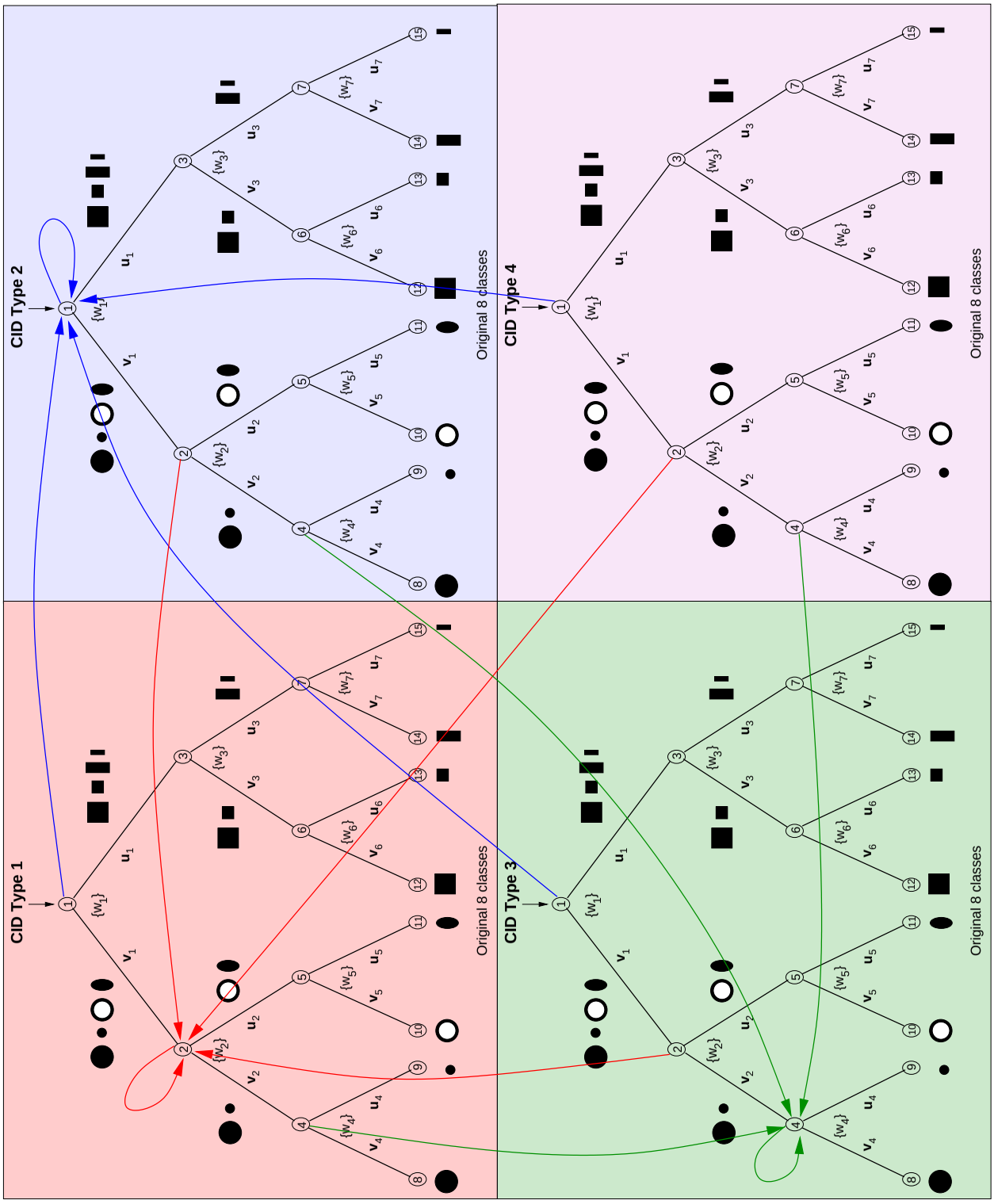


Figure 2: Illustration of the hypertree idea.

3 Classifier Parameter Specification and Training

In this section we review the manner in which BTCs and BHCs are specified starting from a training data set and ending with an operational classifier.

3.1 Tree Topology

The tree topology is the set of connections between the tree nodes. The tree can be balanced, as in Figure 3, or unbalanced, as in Figure 1. The best topology for a particular classification problem is dependent on the statistical nature of the problem. In general, it is advantageous to determine the topology in some adaptive manner that employs measurements on the training data. Analysis of balanced and unbalanced BTCs is reported in [1].

The tree topologies are specified using an integer-valued vector, whose structure is explained here. Each successive pair of integers in the vector specify the sizes of the left and right superclass sets for a node in the tree. The convention used to create the vector is to start at the root node and take the left path out of that node to find the next node. Record the two integers specifying that node's superclass sizes and continue taking the left path out of each node until a terminal node is reached, then travel back up the tree, taking the first available right-going path that is found. This is repeated until all decision nodes are specified in the vector. The balanced tree of Figure 3, for example, is specified by

$$\mathbf{T} = [44221111221111],$$

and the unbalanced tree of Figure 1 is specified by

$$\mathbf{T} = [53141321111211].$$

3.2 Superclass Selection

For a fixed topology $\mathbf{T}$, there are many choices for the sets of left and right superclasses denoted by $\{L_i\}$ and $\{R_i\}$, respectively. The superclasses may be specified in advance, which would also implicitly define the topology. This may be reasonable when certain class subsets are obviously related in some manner, such as those classes whose images consist of arcs and those that consist of straight edges. But in most cases, it will not be obvious how to specify the best topology or superclass sets, and so we'll need an algorithm that finds them automatically during training.

3.3 Feature-Vector Specification

If the topology and superclasses are specified, then there are two remaining classifier parameters to specify: the measurement vectors and the average feature vectors. The measurement vector is determined by using the *local discriminant basis* (LDB) [5] at each decision node [3, 1]. This algorithm finds the best wavelet basis (using the specified mother wavelet type) for representing the classes for the purpose of *discriminating between classes*. This operation is analogous to the more familiar *best-basis algorithm* for finding the best wavelet basis for representing the classes for the purpose of *compression*. The feature-vector specification algorithm then finds the best K vectors in the LDB. These K wavelet basis vectors are then used as the measurement vector.

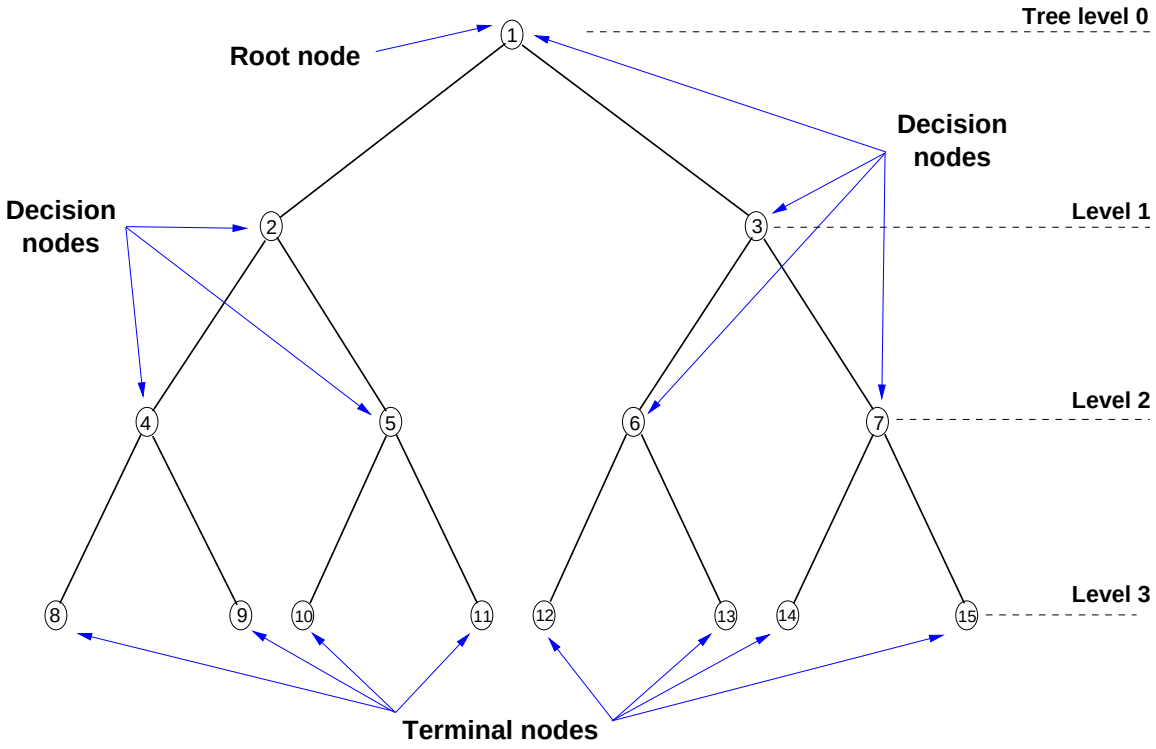


Figure 3: A balanced tree.

Finally, the algorithm uses the obtained measurement vector to find the average feature values for the two involved superclasses. During operation, the measurement vector is applied to the data to be classified, and the resultant K-vector is compared to the left and right average feature vectors.

3.4 Joint Tree Topology, Superclass Selection, and Feature-Vector Specification

The key algorithm in this work is the algorithm for joint specification of the topology, superclasses, and feature vectors for a generic BTC. We have not proved that this algorithm produces the optimal classifier, but our experiments show that the algorithm is adept at automatically determining appropriate topologies, superclasses, and features, and that performance can be quite good. The difficulty with finding the optimal set of tree parameters is computational. For problems with even a modest number of classes C, the number of topologies times the number of superclass selections is very large. This is further compounded when there is more than one sensor modality, so that the number of CIDs M is greater than one.

The joint tree-specification algorithm is not optimal because it does not examine all possible combinations of tree topology and superclass selection. Instead, it assumes that good superclass selections can be made for each node by properly modifying a previously chosen superclass split. In particular, the algorithm first finds the best split that puts one class in L and the remaining classes in R. By "best split" we mean the choice resulting in lowest ambiguity. Then the algorithm finds the best choice for adding one of the elements of R to L, and so on. The detailed algorithm statement is shown in Figure 4.

1. Obtain a set of training data for the C-class problem of interest.
2. Select a specific CID type.
3. Initialize a BTC structure:
   (a) Specify the wavelet type (e.g., Haar, Daubechies, Symmlet).
   (b) Specify the wavelet-packet tree depth J.
   (c) Specify the feature-vector length K.
4. Start with the root node, node $n = 1$.
5. Denote by P_n the inherited set of classes for node n. Let P_n have size N.
6. Set $a_{min} = 1.0$.
7. Let $L_0 = \{\}$, $R_0 = P_n$, $i = 0$.
8. Denote the size of R_i as P. Let $L_n(j) = \{j, L_i\}$ and $R_n(j) = P_n - L_n(j)$.
9. Compute LDB for node n.
10. Select best K elements of LDB to form $\mathbf{w}_n$.
11. Compute $\mathbf{u}_n$ and $\mathbf{v}_n$.
12. Compute the ambiguity $a_n(j)$.
13. Repeat Steps 8–12 until $j = P$.
14. Increment i, decrement P.
15. Retain the $L_n(j)$ set for which $a_n(j)$ is minimum. Denote best $L_n(j)$ by L_i.
16. If the minimum ambiguity $a_n(j)$ is greater than a_{min}, then goto Step 18, else set $a_{min} = a_n(j)$.
17. Repeat Steps 8–15 until L_i has size greater than or equal to $N/2$.
18. Find childless decision node with specified parent node; denote by n if it exists and goto Step 5.
19. Stop.

Figure 4: Automatic specification of BTC topology, superclasses, and features.

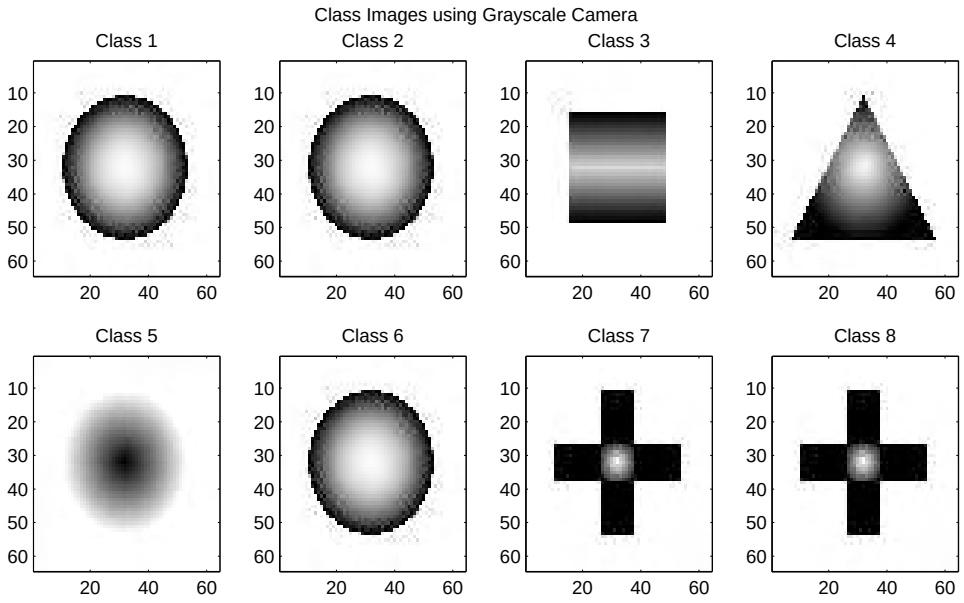


Figure 5: An eight-class problem used for illustration.

Example of Joint Algorithm in Action.

As an example, consider the eight-class problem shown in Figure 5. In this example, which is explored in more detail in Section 5, the eight classes are interpreted as objects viewed through a gray-scale camera. This operation results in perfectly ambiguous class subsets, such as $\{1, 2, 6\}$, and $\{7, 8\}$. The automatically obtained BTC for this CID is shown in Figure 6. Each decision node is annotated with its computed ambiguity and each terminal node is annotated with a class label. The important point here is that the automatic algorithm perfectly captures the inherent structure of the problem: 7 will be mistaken for 8 and vice versa, and 1, 2, and 6 will be confused. This simply reflects the strong ambiguities in the problem.

3.5 Specifying the BHC

A BHC is constructed from two or more constituent BTCs. The BTCs must attack either the same classification problem or subsets of a single problem, but otherwise can have different topologies, CIDs, superclass selections, wavelets, feature lengths K, etc. Crucial to creation of a BHC is the notion of *corresponding nodes*. For convenience, the mathematical definition of corresponding nodes is repeated here [1]:

Definition 1 (Corresponding Nodes) *Let T_1 and T_2 denote two distinct BTCs and let n_1 and n_2 denote decision nodes from T_1 and T_2, respectively. If the union of the left and right superclasses for nodes n_1 and n_2 match, then these two nodes are* corresponding. *If the superclasses are equal, then the nodes are* equivalent. *Note that the binary-tree parameters C_1 and C_2 for T_1 and T_2 need not be equal.* ■

Let H denote a binary hypertree with N constituent binary tree classifiers each with parameter C and each addressing the same classification problem. Assign the node pointers for each node

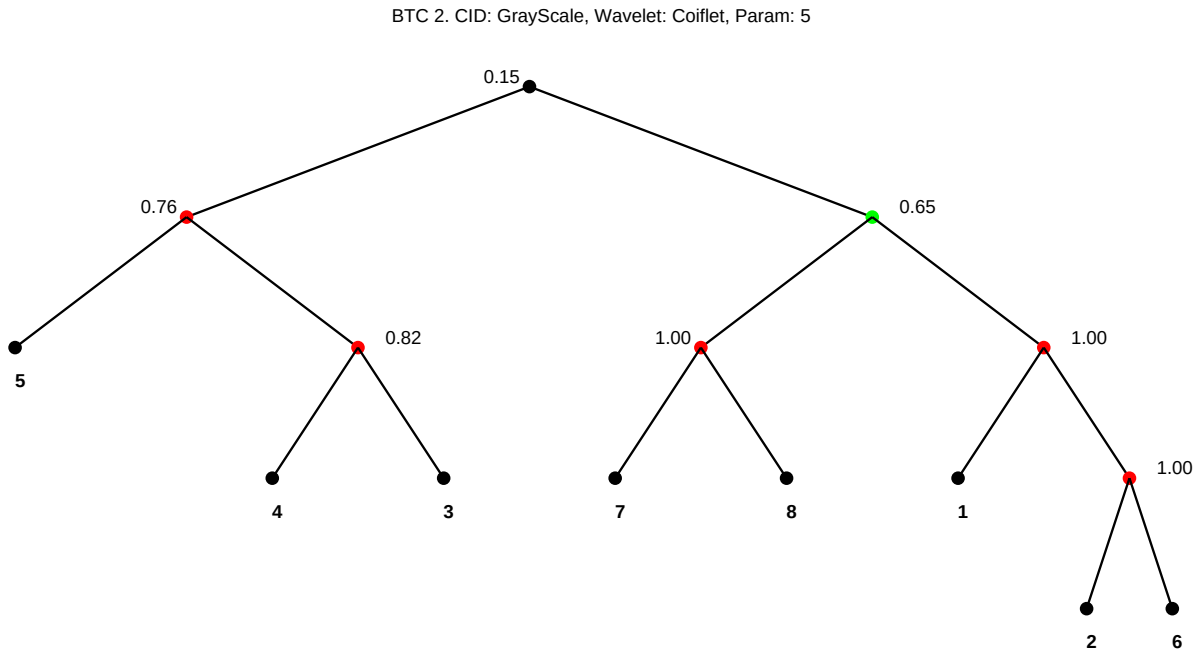


Figure 6: A BTC automatically obtained and plotted using the developed software.

such that the node points to the BTC with minimum ambiguity-corresponding node. These pointers will be used during BHC operation.

4 Classifier Operation

The operation of the three classifier types is described in this section. Detailed algorithm statements for classifier operation are provided in the companion report [1]. Here the operation is described in a less formal way.

4.1 BTC and BSC

The operation of these classifiers is particularly simple. First, the item to be classified is obtained. This is a single CID for the problem of interest for the BTC and all available CIDs for the BSC. Starting at the root node of the BTC (or BSC), the wavelet signal processing operations encoded in the measurement vector $\mathbf{w}$ are used to compute the feature vector $\mathbf{f}$, which has length K. The correlation coefficient between $\mathbf{f}$ and the left-going average feature vector $\mathbf{v}$ is computed and compared to that for the right-going feature vector $\mathbf{u}$. Take the left path if the feature is more highly correlated with $\mathbf{v}$, otherwise take the right path. Continue in this manner until a terminal node is reached. The class label associated with the reached terminal node is the decision.

Note that the use of the correlation coefficient means that the amplitude of the data element to be classified is irrelevant, since it is normalized away.

4.2 BHC

The BHC operation is only slightly more complex than operation for the BTC. First, the item to be classified is obtained. Start with whatever CID is cheapest to obtain or start with the CID corresponding to the BTC with minimum-ambiguity root node. Compute the required two correlation coefficients and select the left or right path out of this first node. Check the hypertree pointer in the new node. If it points to the current tree and current node, then continue, else jump to the tree and node stored in the pointer. The BTC that is jumped to may correspond to the same CID or a new CID. If it is a new CID, perform the operations needed to obtain the data and compute the required correlation coefficients. Continue in this way until a terminal node is reached.

4.3 Path Correction in the BTC

It will be very rare to construct a tree classifier such that all decision nodes have very small or zero ambiguity. This will happen only for problems that are inherently easy to solve. Therefore, BTCs for real-world difficult problems will have one or more nodes with relatively high ambiguity and, therefore, non-negligible probability of decision error. Because the computational operations required during tree-traversal are modest in complexity, we consider multiple traversals of the tree in an attempt to provide the best possible decision. In particular, we outfit the classifier with the capbability of detecting a poor-quality decision, and a means to avoid that decision during a subsequent traversal. We call this notion *path correction*.

The key idea in path correction is that of *decision quality.* When the node ambiguities encountered during a particular tree traversal are all small, we would expect that the winning correlation coefficient at each node in the traversed path will be close to one. On the other hand, if one of the encountered nodes has high ambiguity and an erroneous decision is made at this node, we would expect that all winning correlation coefficients for the remaining nodes in the path will be far from one since the path cannot contain the true class. Therefore, it should be possible to detect, based on the sequence of winning correlation coefficients obtained during traversal, a “bad path.” Then the tree can be retraversed with the detected low-quality decision ruled out.

In our path-correction algorithm, the decision quality is simply the value of the winning correlation coefficient at the decision node just above the chosen terminal node. The tree is traversed as many times as needed, until either the decision quality is high, or the root node is reached. If this happens, the path correction algorithm simply produces the original decision.

We will see that path correction is particularly well suited to situations involving weak ambiguities. For problems containing strong ambiguities, path correction should not help since multiple paths through the tree will result in identical decision qualities (cf. Figures 5 and 6).

5 Experiments

In this section, we report on various experiments aimed at validating our classification theory and design. In Section 5.1, we examine a one-dimensional problem involving 16 classes and three CID types. The classes correspond to the 16 unique maximal-length shift-register (MLSR) sequences for shift-register length eight [8]. In Section 5.2, we examine a synthetic two-dimensional problem involving eight classes and four CID types. The classes correspond to eight physical objects seen

through four camera types and involve serious ambiguities for each CID. In Section 5.3 we look at several classification problems using publicly available data. The involved classes include DNA sequences, letters of the roman alphabet, and data related to the space shuttle.

The goals of the experiments are as follows:

1. Validate the operation of the basic binary tree classifier.
2. Validate the performance ordering BSC $\geq$ BHC $>$ {BTC}.
3. Determine the data-burden advantage of the hypertree classifier over the supertree (fusion) classifier.
4. Determine the influence of the particular choice of wavelet.
5. Determine the efficacy of path correction for the BTC and BSC.
6. Determine how well the system *automatically* determines the problem structure; that is, how well the system identifies and isolates ambiguous classes in the produced trees.

5.1 Toy Problem One: One-Dimensional Inputs

In this section, we report on a number of related experiments in which the classes of interest consist of a set of sixteen binary sequences. The sequences have length 256 and are the sixteen unique maximum-length shift-register (MLSR) sequences associated with a shift-register length of eight [8]. The goal of the classification system is to efficiently use the three classifier input data (CID) types to correctly determine which MLSR sequence gave rise to the data.

Classifier Input Data Types.
There are three CIDs for the one-dimensional problem. These correspond to the sequences themselves, filtered sequences, and an auxilliary input called *metal*. Graphs of the sixteen noise-free elements of each CID set are shown in Figures 7–9. The concept here is that one CID is a high-resolution version of the sequences (which will later suffer from low SNR), one is a low-resolution version (which may be less costly to obtain), and the final CID is an auxilliary piece of information that has low discrimination capability, but may be useful for resolving ambiguities arising from the first two CIDs.

Experiment Outputs.
For each distinct subexperiment, the following results will be provided.

1. The obtained classification trees will be displayed. The decision nodes will be annotated with their ambiguity and the terminal nodes with their class label.
2. The overall probability of correct classification (P_{cc}) will be graphed versus classifier index. This graph reveals the overall relations between the BSC, BHC, and the constituent BTCs. We would expect that the BHC and BSC have the best performance, and could be substantially better than any of the BTCs.
3. The confusion matrix will be displayed so that the basic misclassification patterns can be observed and correlated with any obvious class ambiguities.

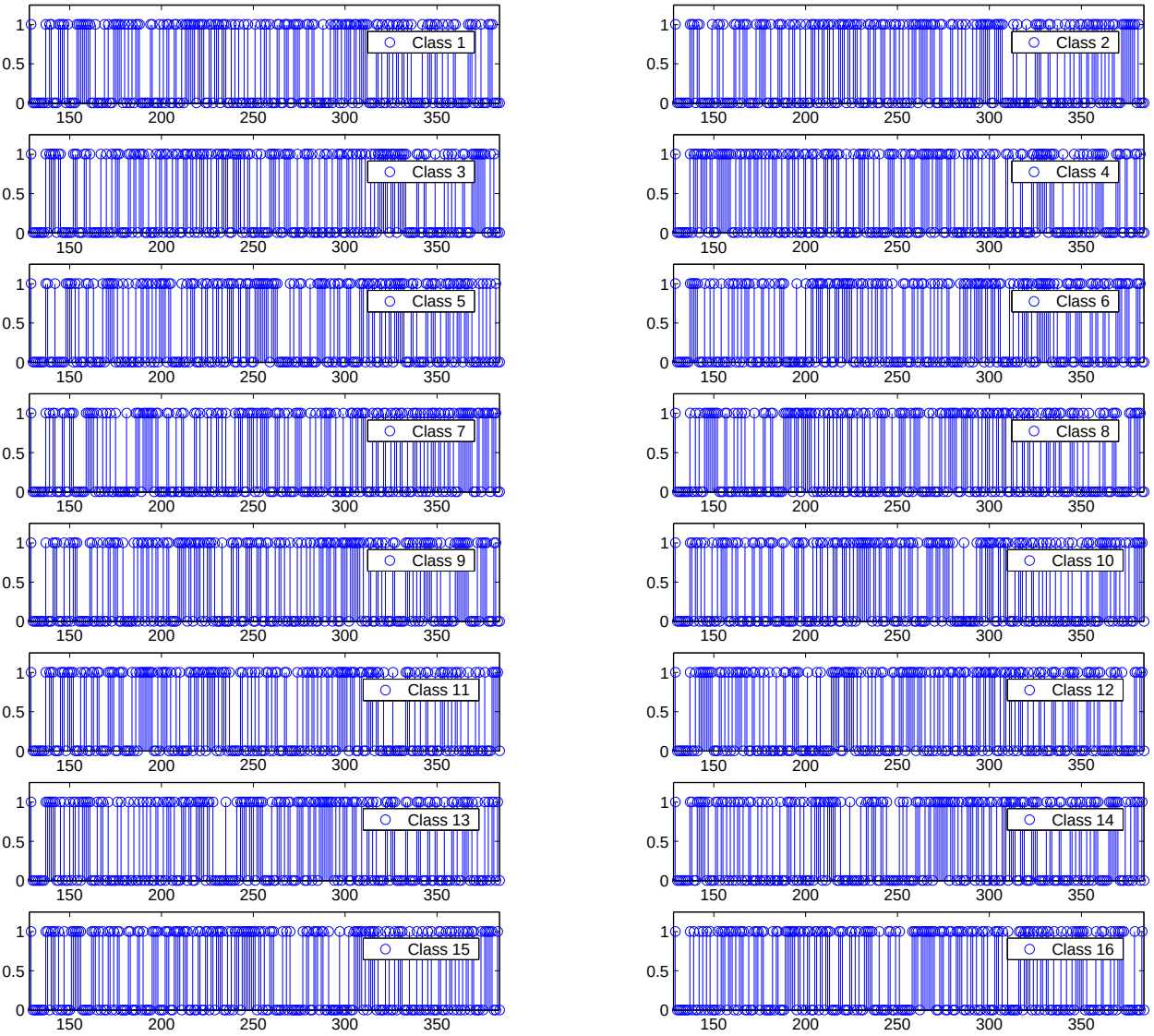


Figure 7: The sixteen MLSR sequences for the one-dimensional experiments.

4. Two histograms will be provided for the quality measure. The first corresponds to all trials for which the classification is correct, and the second corresponds to all other trials. This will allow a quantitative assessment of the quality measure as a tool for detecting incorrect classifier outputs.
5. For the BHC, the average number of required distinct CID types will be reported. This number will be compared to the data input requirement for the appropriate BSC to determine the BHC's average potential savings in required input data.

5.1.1 Experiment 1.1: Basic 1-D Processing

In this first MLSR experiment, BTC, BHC, and BSC structures are obtained for a single wavelet type. Since there are three CIDs and one measurement type, there are three basic BTCs, one BHC,

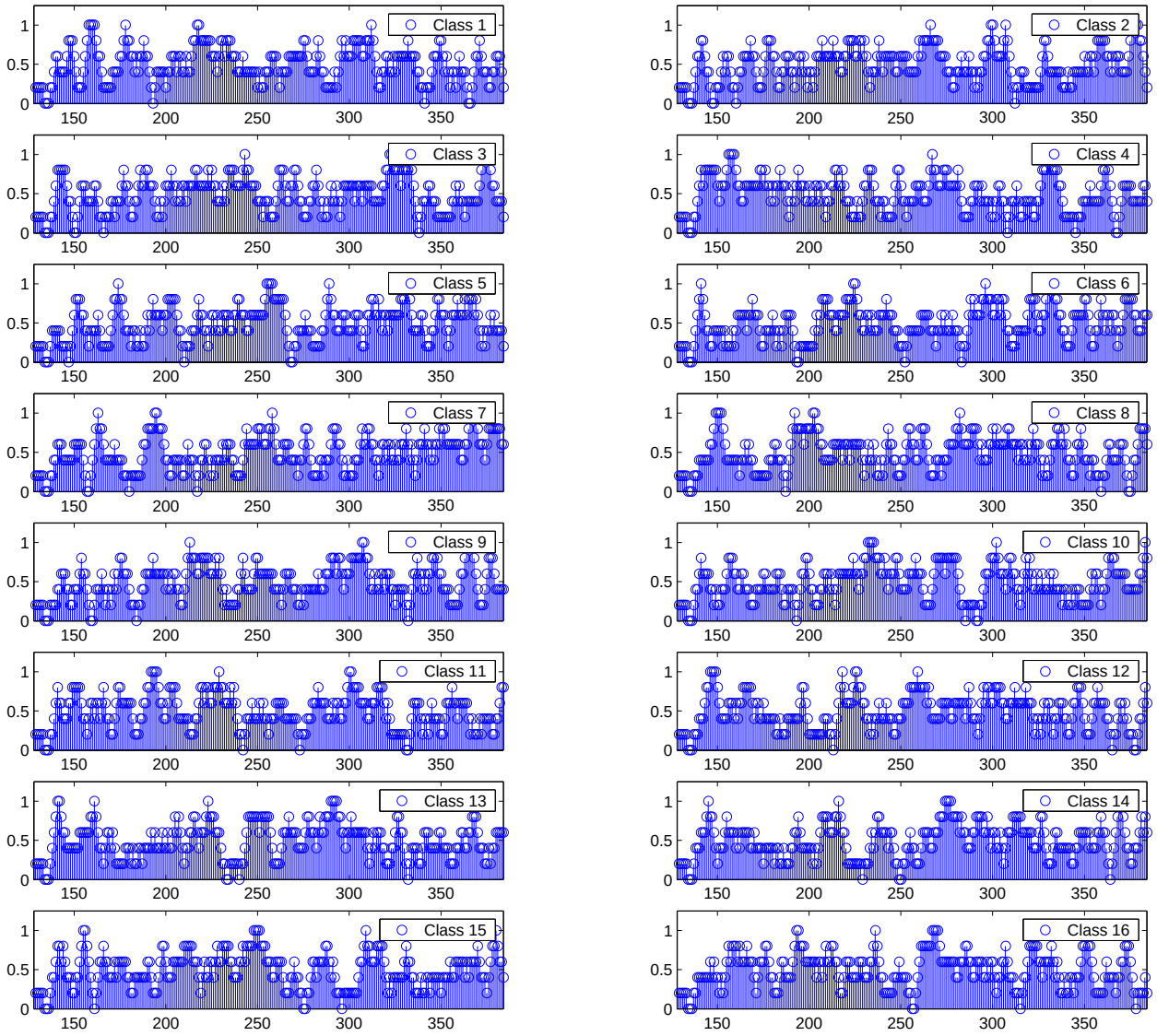


Figure 8: The sixteen filtered MLSR sequences.

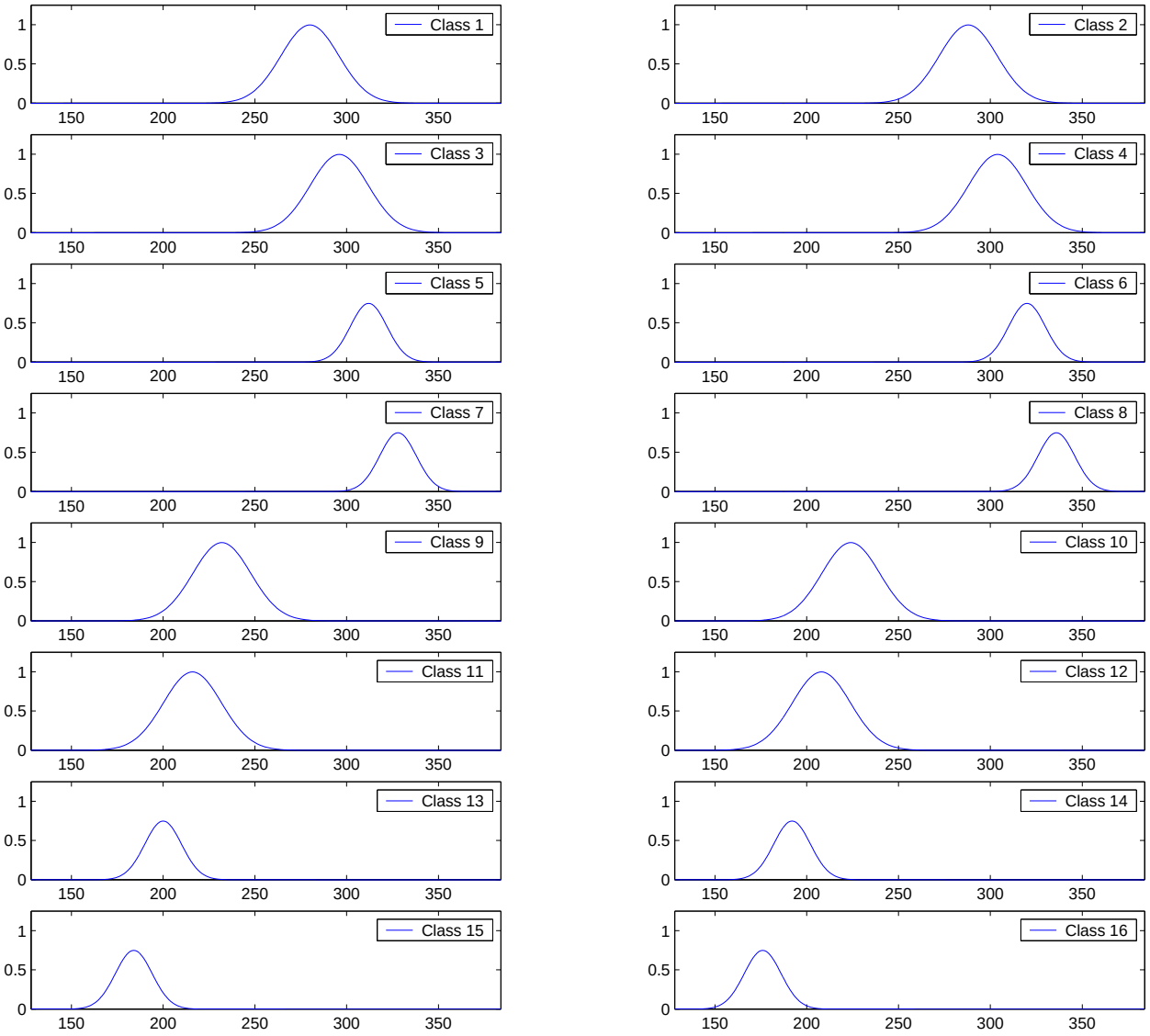


Figure 9: The auxilliary information CID for the one-dimensional experiments.

and one BSC, for a total of five distinct classifiers. Six additional derived classifiers, explained below, are added to obtain eleven classifiers. The experimental parameters are provided in Table 1.

Parameter	Value
Wavelet Type	Coiflet
Wavelet Parameter	5
Feature Length K	20
Number of Classes C	16
BTC/BHC Wavelet Tree Depth J	6
BSC Wavelet Tree Depth J	8
Number of CIDs	3
Data Dimension	[1 256]
Processed Data Dimension	[1 512]
Training SNR	∞
Input SNR CIDs 1,2,3	10, 13, 10dB
Random Translation	None
Random Scaling	None
Tree Topology	Free
Superclass Assignment	Free
Number of Trials	100

Table 1: Experimental parameters for the first 1-D experiment.

Probability of Correct Classification The overall classification performance for the three basic BTCs, the BHC, and the BSC is summarized in Figure 10.

Automatically Obtained BTCs and BSC The obtained BTC structures are plotted in Figures 11–15. The first three of these correspond to the BTCs that are found for the three CIDs, and are called the *basic* BTCs. The remaining six BTCs are formed by fixing the structure of one of the basic BTCs and replacing the CID with one of the other two CIDs, and are called *derived* BTCs. The obtained BSC tree is shown in Figure 15. In these plotted trees, and in all others in this report, nodes that are "jump-to" nodes in the BHC are colored green, and nodes that have high ambiguity are colored red. Green takes precedence over red. Otherwise, the node is black.

Notice the prevalence of low-ambiguity nodes in the BTC for CID 1 (the sequences themselves) in Figure 11. This correlates well with the performance for this BTC shown in Figure 10. For the second CID, we see from Figure 12 that the tree is much more balanced, but that the ambiguities are generally larger. These two effects tend to oppose each other, and performance is as good as for CID 1. For CID 3, the most ambiguous of the three CIDs, we see from Figure 12 that the obtained tree is highly unbalanced and contains some large ambiguities high in the tree. This correlates well with the observed probability of correct classification of 0.4 in Figure 10.

Regarding the BSC in Figure 15, we see that the obtained tree is distinct from any of the three basic BTCs and that there are no high-ambiguity nodes in the upper parts of the tree. However, the tree is severely unbalanced at the root node, and this can cause a serious performance shortfall.

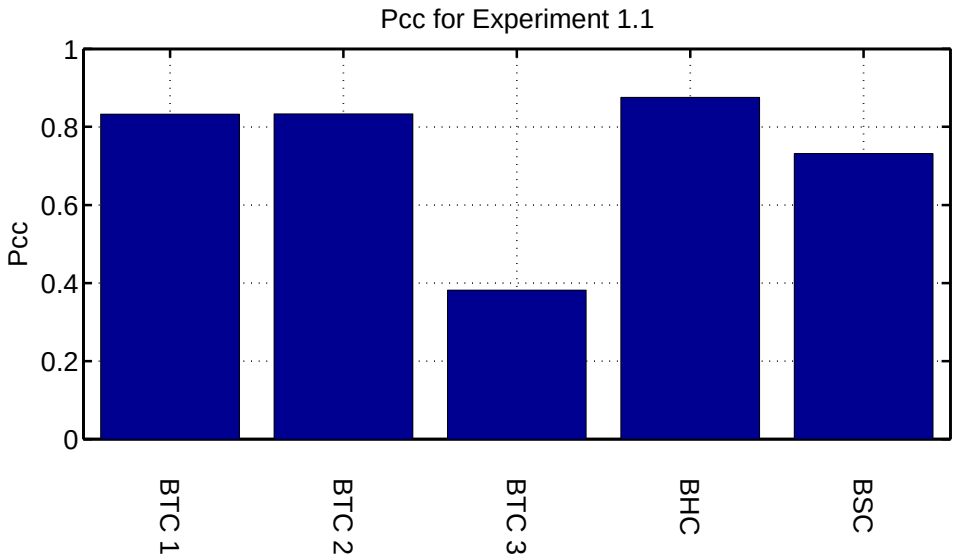


Figure 10: Classification performance for Experiment 1.1

Confusion Matrices The confusion matrices for the three basic BTCs, the BHC, and the BSC are shown in Figures 16–18, respectively. Note the correlation between the appearance of the confusion matrix and the structure of the trees. For example, for CID 3, we have the tree in Figure 12 and the confusion matrix in Figure 17. From the latter, we see that classes 5 and 6 are commonly confused for a large number of input classes. From the tree, we see that these two classes are the only possible decisions when taking the left path out of the root node.

Quality Measures The histograms of the quality measures for Experiment 1.1 are shown in Figures 19–23.

Conclusions for Experiment 1.1

1. The BSC is outperformed by both the BTCs and the BHC. Since the BSC has access to all available input data, it should not be outperformed by any of the constituent BTCs, and its performance should upper-bound that of the BHC. The reason for this result is unclear.
2. The BHC outperforms the BTCs. This basically means that the algorithm for forming the BHC is working well, choosing the best nodes in the correct trees so that switching from one BTC to the next results in a performance improvement on the average.
3. The algorithm for jointly choosing tree topology and feature-vector values is generating a variety of topologies and tree-nodes with small ambiguities. We also observe that the algorithm is doing a good job in pushing large ambiguities downward in the tree, which is a necessary condition for good performance. However, since the structure of the individual class elements is hard to discern (each of the classes is represented by an essentially random binary string), it is difficult to determine if the tree topologies make sense in terms of good splits of the incoming class labels into two sets of outgoing labels. This task is made much easier in Experiment 2, for which the structure of the data is more easily grasped visually.

4. The quality measure is a good indicator of the correctness of a classification decision. A large majority of incorrect decisions result in a quality measure that is less than 0.90, while virtually all correct decisions result in a quality measure that is greater than 0.95. This implies that the BTCs and the BSC may benefit from *path correction* (see Sections 4 and 5.1.3).

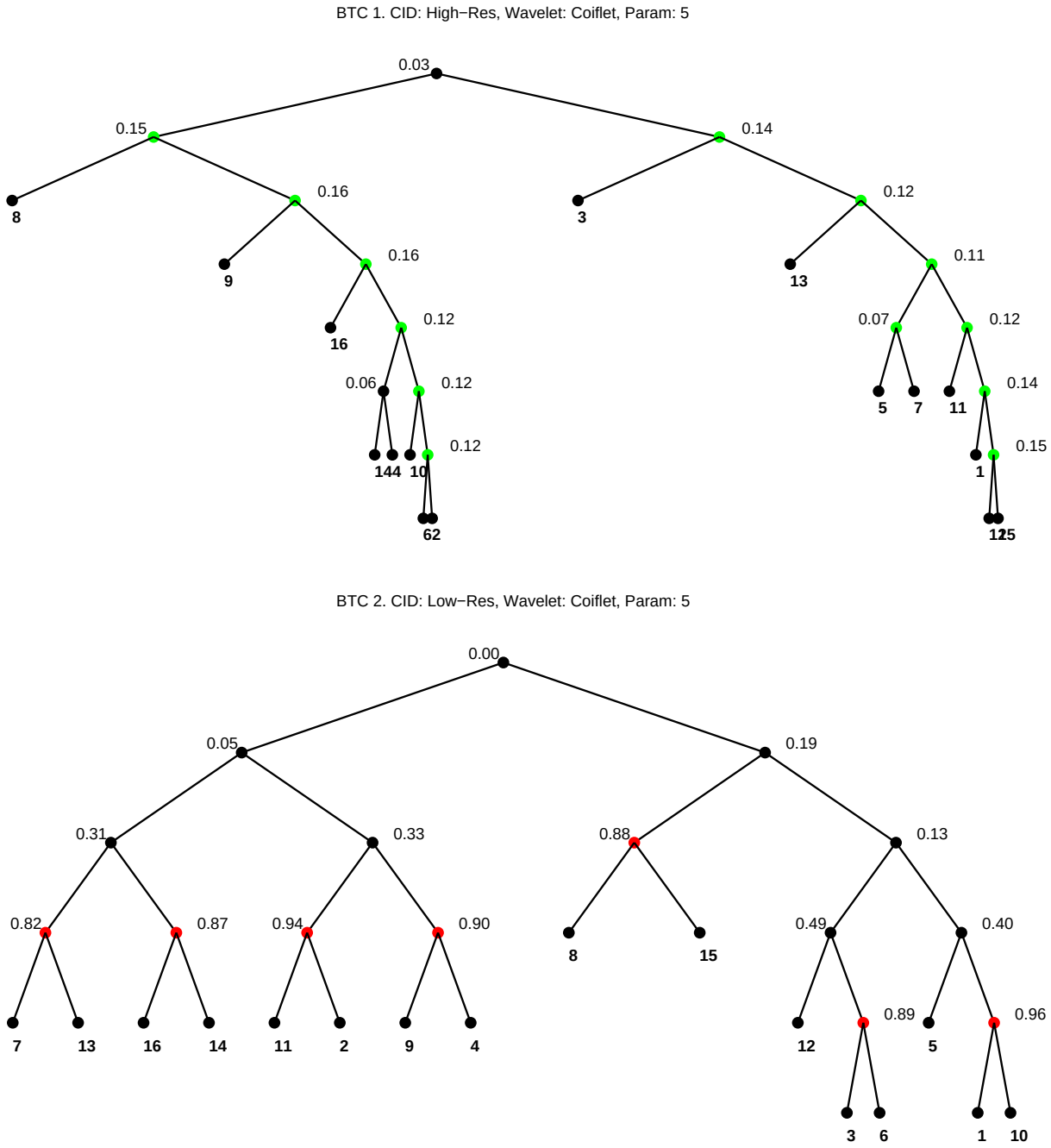


Figure 11: BTCs obtained for Experiment 1 (1–2 of 9).

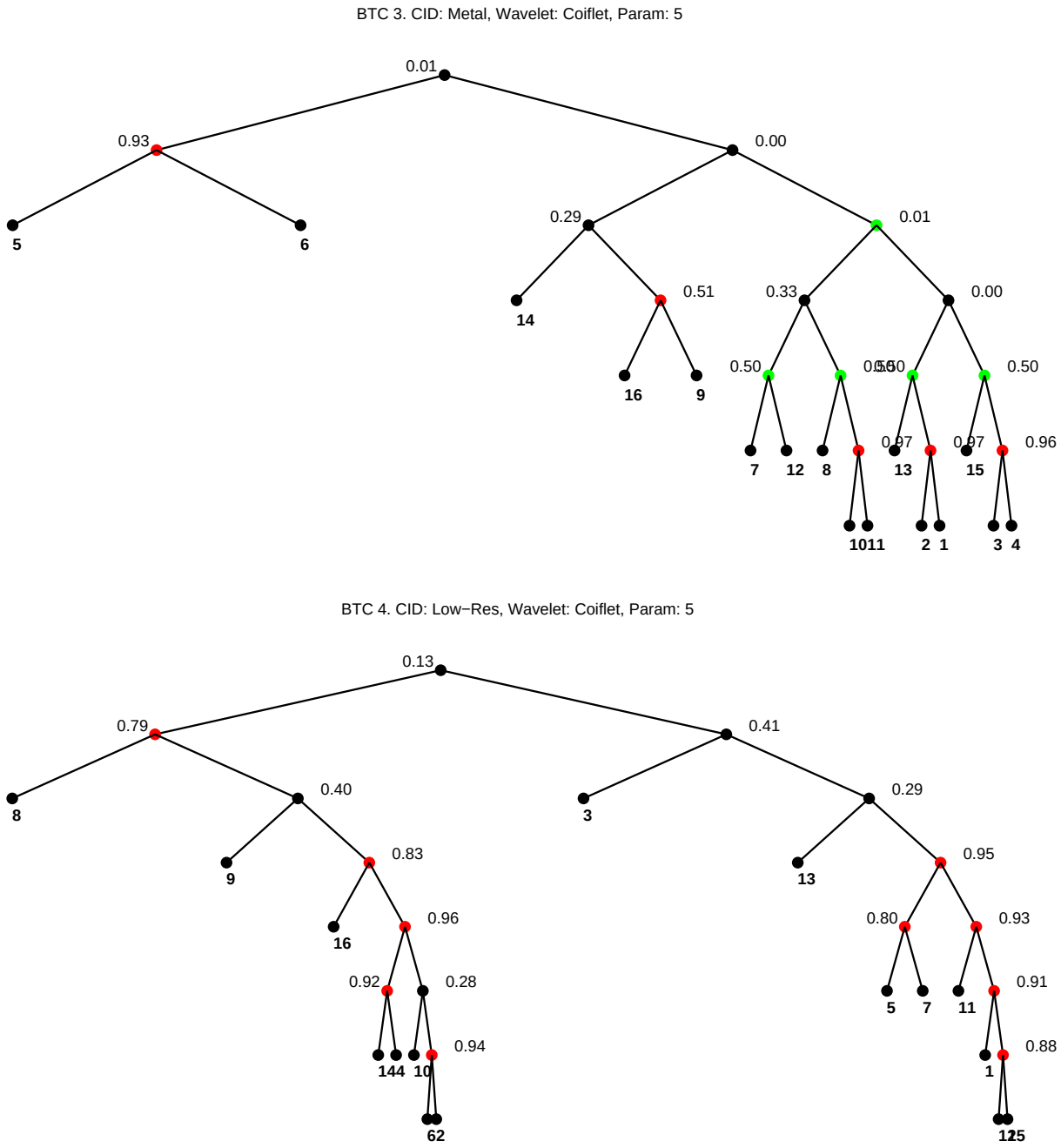


Figure 12: BTCs obtained for Experiment 1 (3–4 of 9).

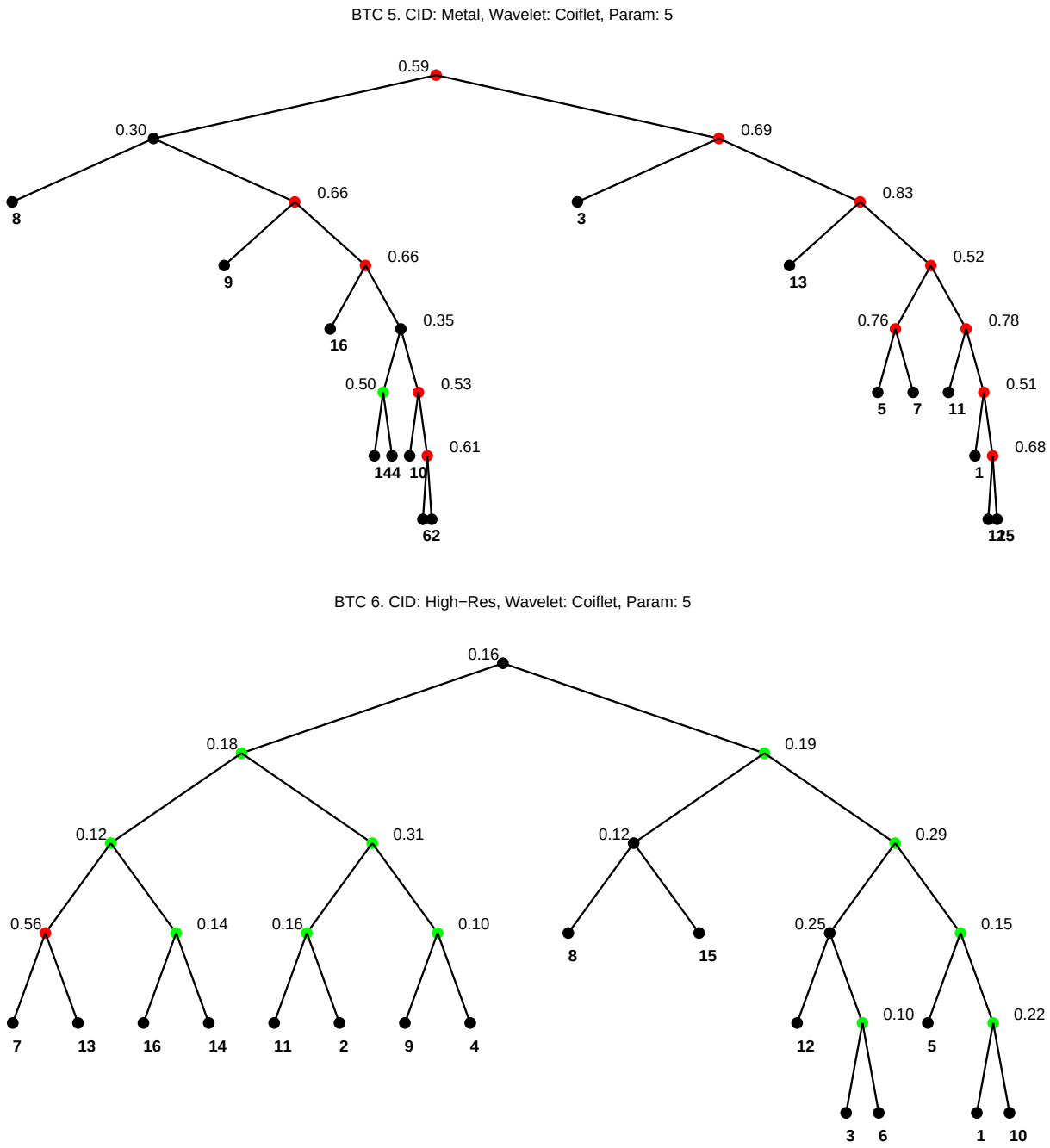


Figure 13: BTCs obtained for Experiment 1 (5–6 of 9).

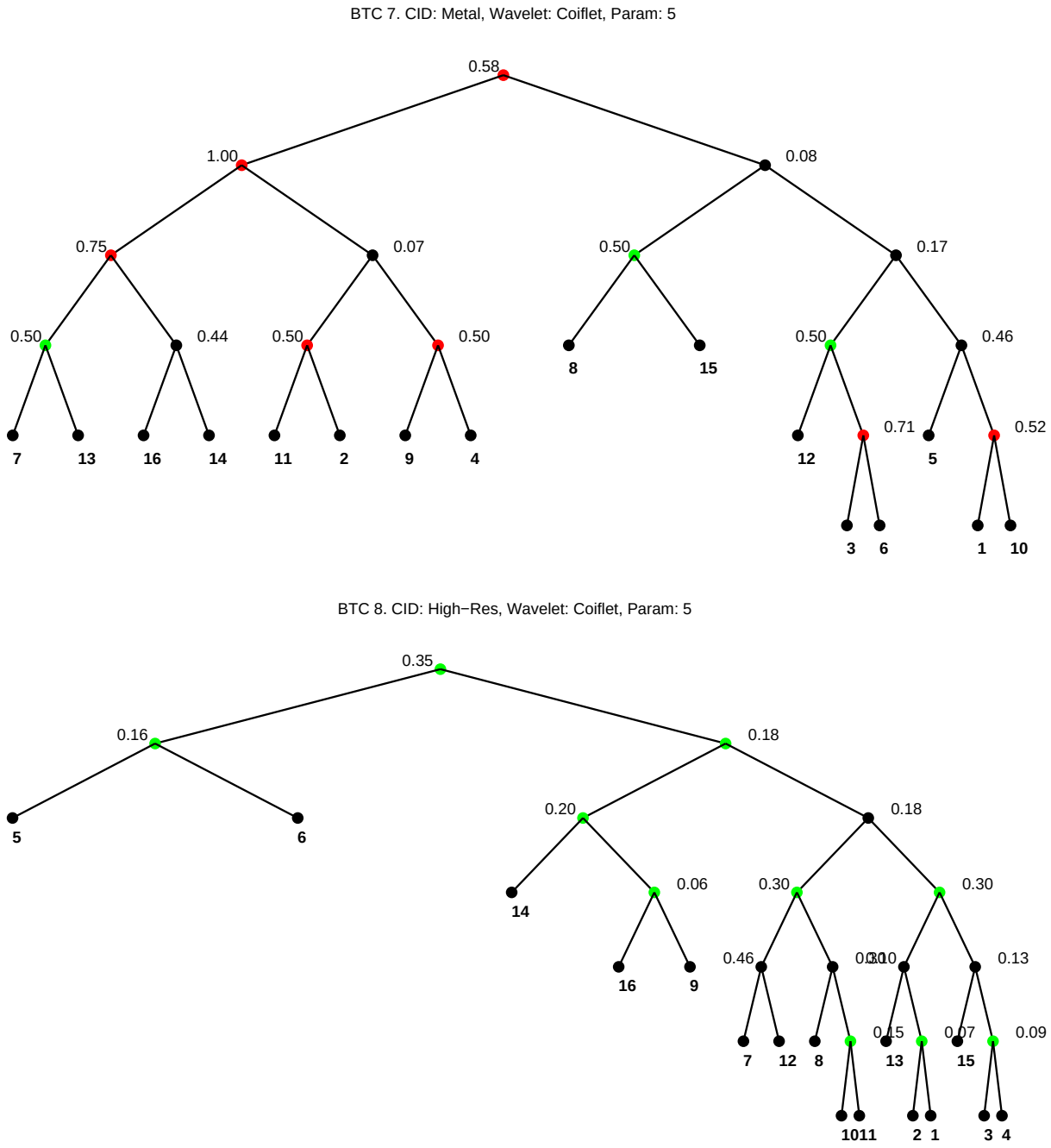


Figure 14: BTCs obtained for Experiment 1 (7–8 of 9).

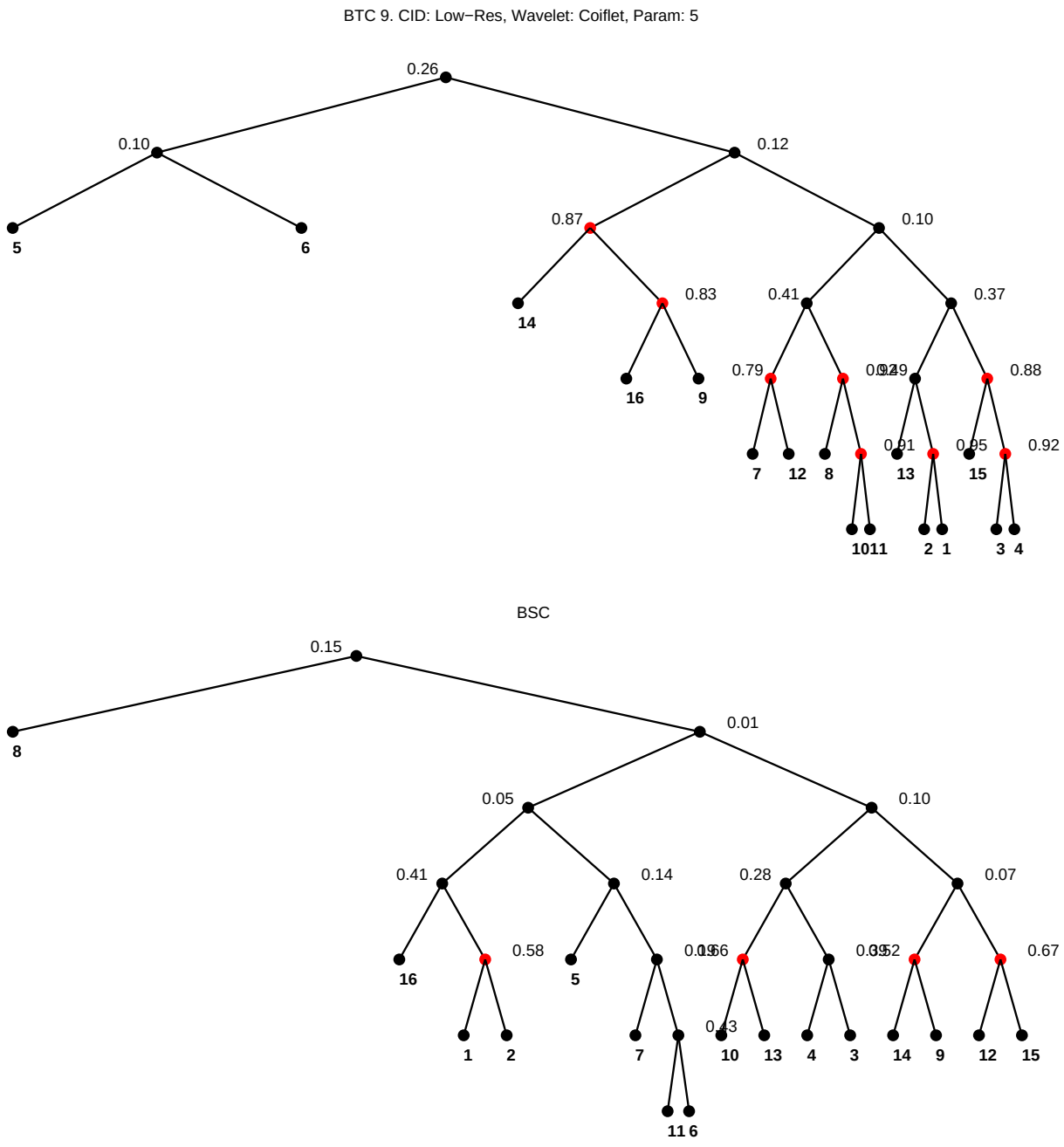


Figure 15: BTC 9 and the BSC obtained for Experiment 1.

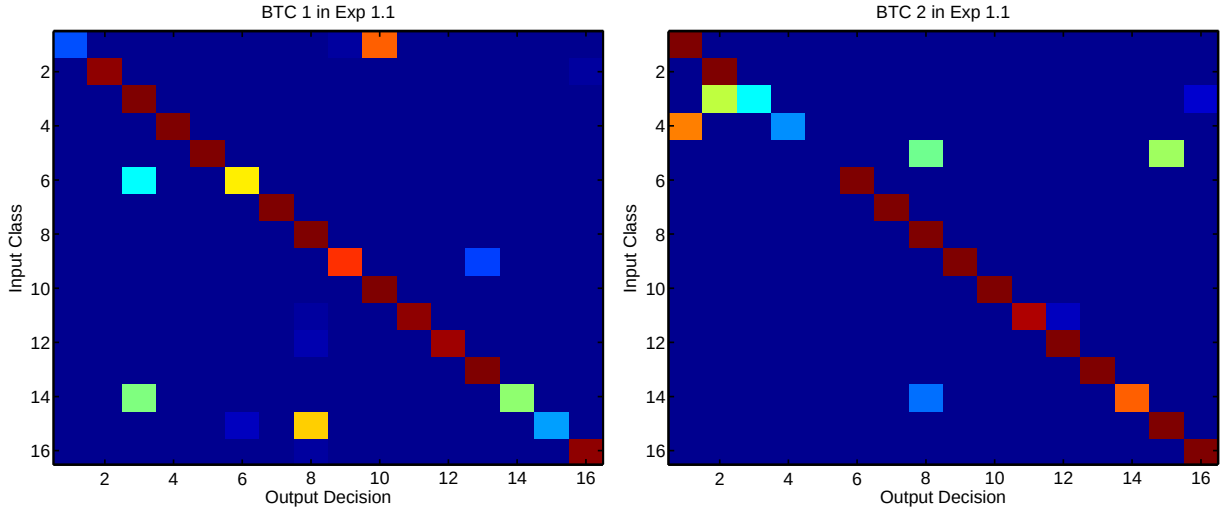


Figure 16: Confusion matrices for BTCs 1 and 2 in Experiment 1.1.

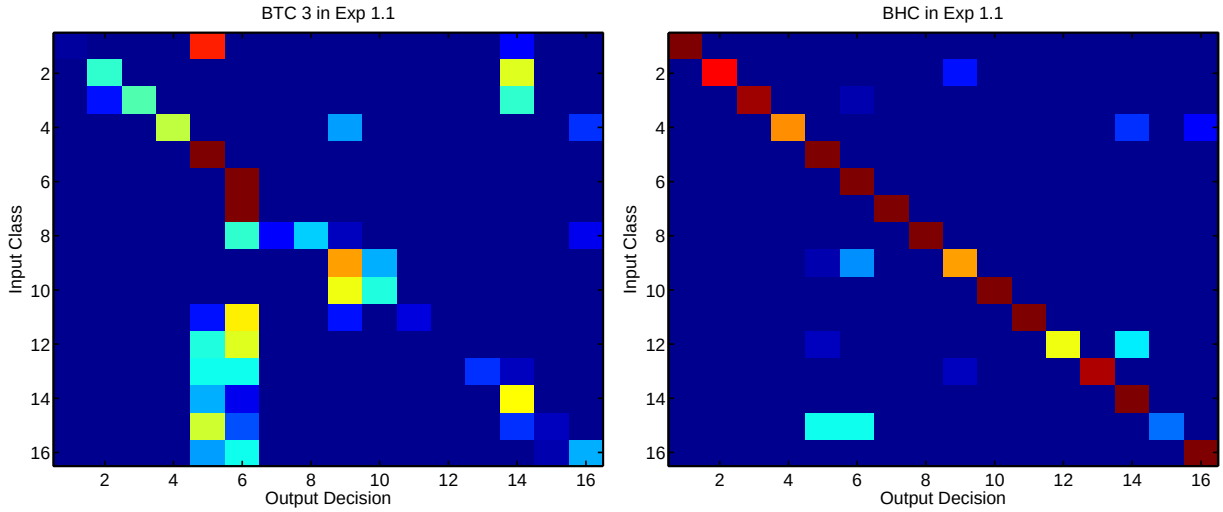


Figure 17: Confusion matrices for BTC 3 and the BHC in Experiment 1.1.

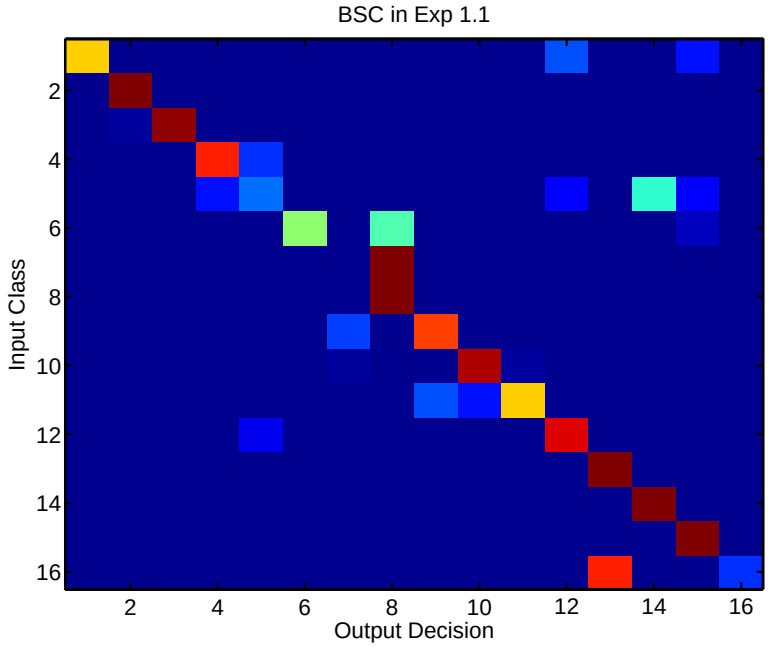


Figure 18: Confusion matrix for the BSC in Experiment 1.1.

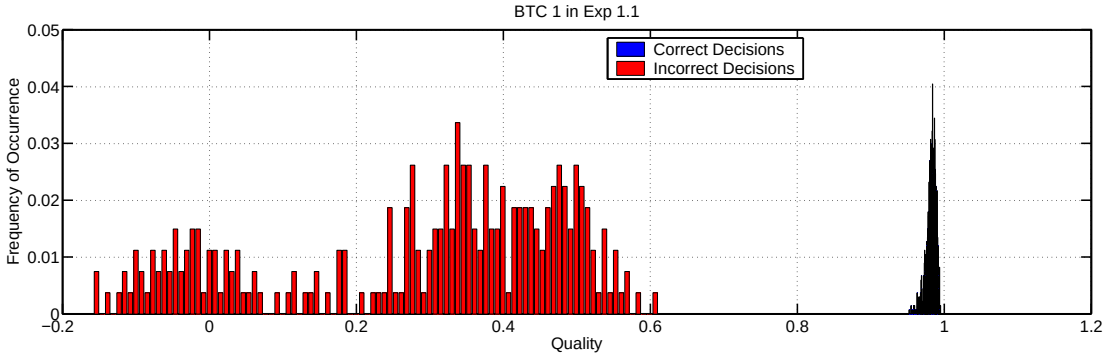


Figure 19: Quality histogram for BTC 1 in Experiment 1.1.

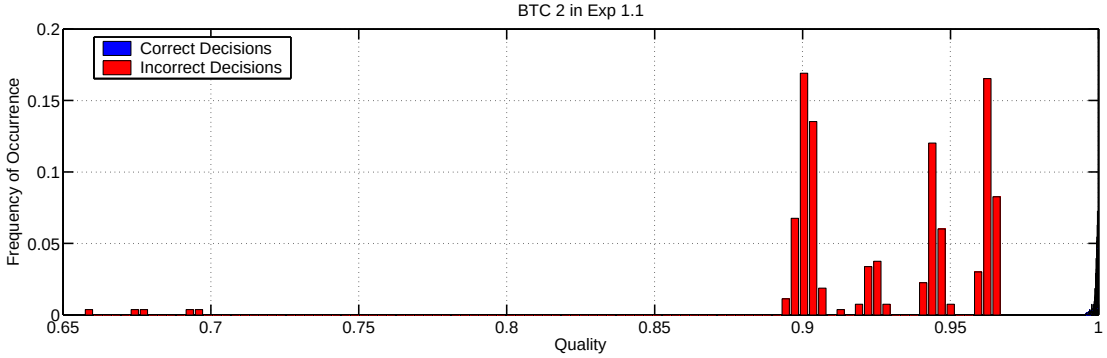


Figure 20: Quality histogram for BTC 2 in Experiment 1.1.

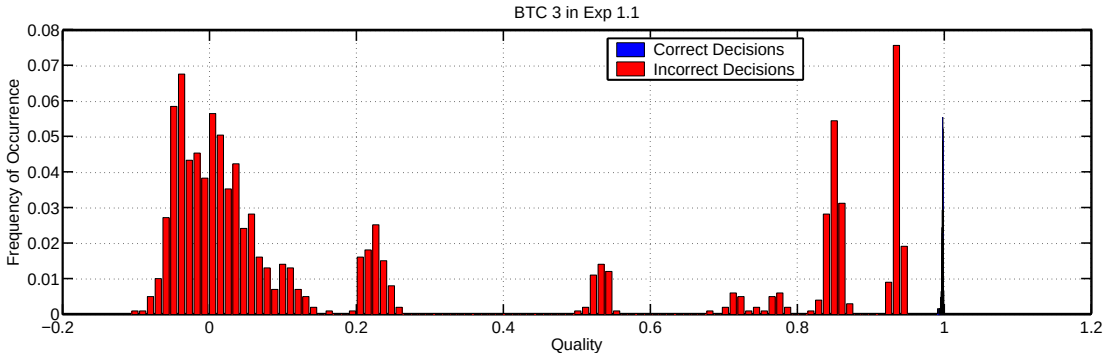


Figure 21: Quality histogram for BTC 3 in Experiment 1.1.

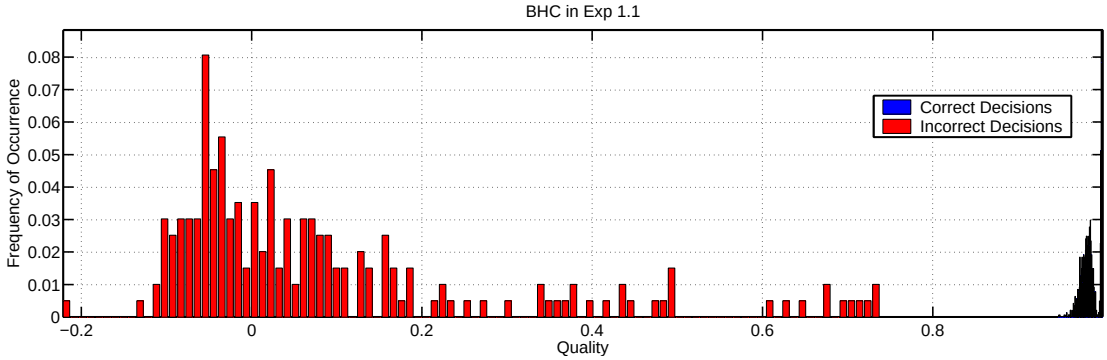


Figure 22: Quality histogram for the BHC in Experiment 1.1.

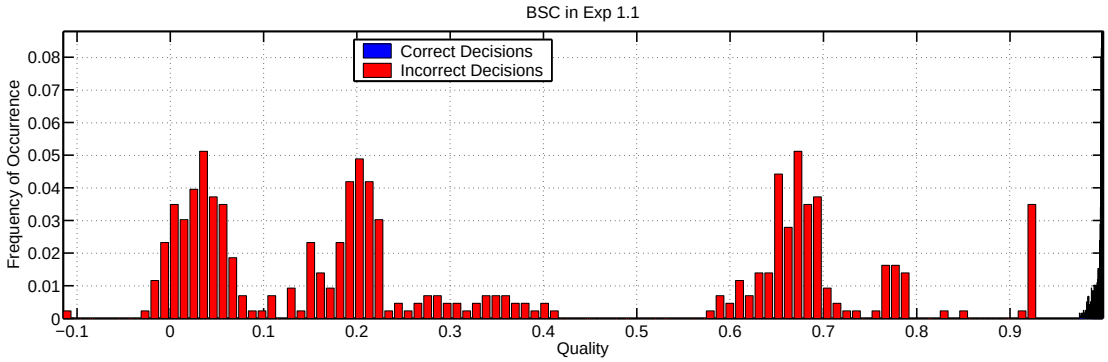


Figure 23: Quality histogram for the BSC in Experiment 1.1.

5.1.2 Experiment 1.2: Multiple Wavelet Types

In this second one-dimensional experiment, the number of constituent BTCs available for use in the BHC is increased by allowing distinct measurement functions (wavelet types) in addition to the distinct data types (CIDs). In particular, ten distinct wavelet types are used instead of just one as in Experiment 1.1. The parameters for Experiment 1.2 are shown in Table 2.

For each wavelet type, the three basic BTCs are found through the algorithm that jointly determines tree topology and feature-vector values. This yields a total of thirty BTCs. For each of these, two additional BTCs are created by fixing the topology and superclass choices, selecting one of the other CID types, and retraining the structure. This yields 60 more BTCs. The BHC is based on these 90 classifiers. Finally, ten BSCs are created, one for each wavelet type.

Parameter	Value
Wavelets	
Beylkin	
Coiflet	1
Coiflet	5
Daubechies	4
Daubechies	20
Symmlet	4
Symmlet	10
Vaidyanathan	
Battle	1
Battle	3
Feature Length K	20
Number of Classes C	16
BTC/BHC Wavelet Tree Depth J	6
BSC Wavelet Tree Depth J	8
Number of CIDs	3
Data Dimension	[1 256]
Processed Data Dimension	[1 512]
Training SNR	∞
Input SNR CIDs 1,2,3	10, 13, 10dB
Random Translation	None
Random Scaling	None
Tree Topology	Free
Superclass Assignment	Free
Number of Trials	100

Table 2: Experimental parameters for the second 1-D experiment.

Automatically Obtained BTCs and BSCs The thirty basic BTCs, the BHC, and the ten BSCs for Experiment 1.2 tend to be either largely balanced or severely unbalanced. For reasons of

brevity, we do not display the trees in this section. The results are fairly similar to those for Experiment 1.1.

Probability of Correct Classification The probabilities of correct classification for the thirty basic BTCs and the ten BSCs in Experiment 1.2 are shown in Figure 24. Since the BHC is outperformed by several BTCs, including those used in Exp 1.1 (BTCs 7–9), the algorithm for constructing a BHC from constituent BTCs is flawed, else it could restrict its attention to only BTCs 7–9 and thereby obtain the performance of the BHC in Experiment 1.1.

Note also that the BTC and BSC performance is strongly dependent on the wavelet. The best wavelet for an individual BTC is Daubechies with CID 2 (BTC 11), and the best wavelet for the BSC is Beylkin (BSC 1).

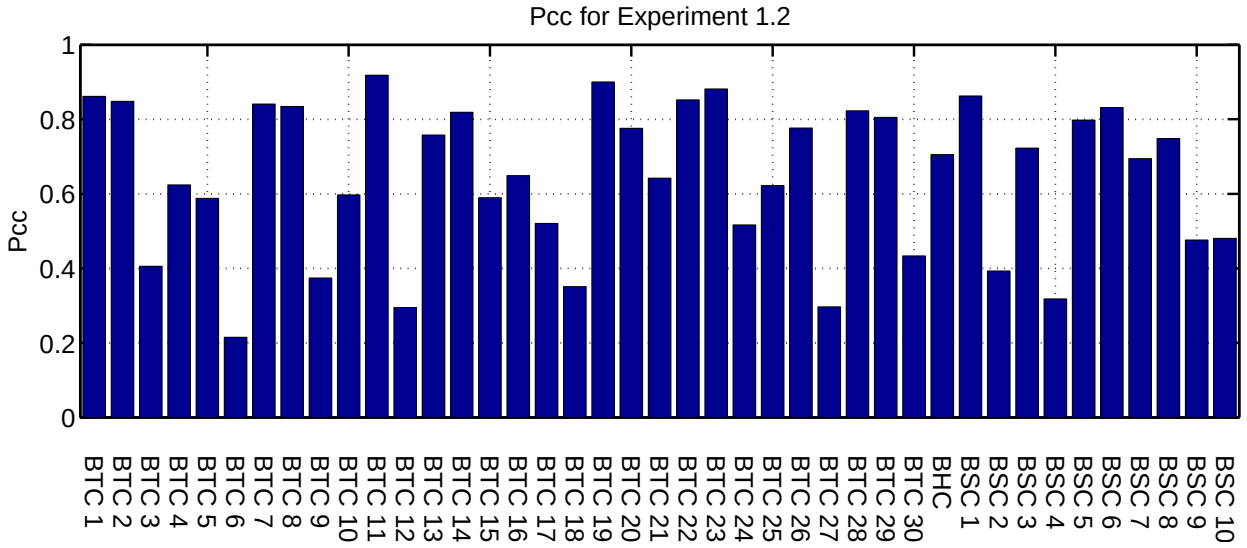


Figure 24: Classification performance for Experiment 1.2

Confusion Matrices The obtained confusion matrices for the BTCs are shown in Figures 25–39, for the hypertree classifier in Figure 40, and for the BSCs in Figures 41–45.

Quality Measures Finally, the quality measures for the various classifiers in Experiment 1.2 are shown in Figures 46–66.

Conclusions for Experiment 1.2

1. Many of the BTCs for this experiment result in good-to-excellent performance, but the exact performance level is dependent on the specific wavelet.
2. The BHC is not correctly constructed from the set of constituent BTCs. Our suspicion is that the algorithm is relying too heavily on the use of ambiguity to choose the "jump-to" nodes in the hypertree. This must be balanced by the power of the obtained average feature vectors (relative to the average power of the input classes). Very weak average features may

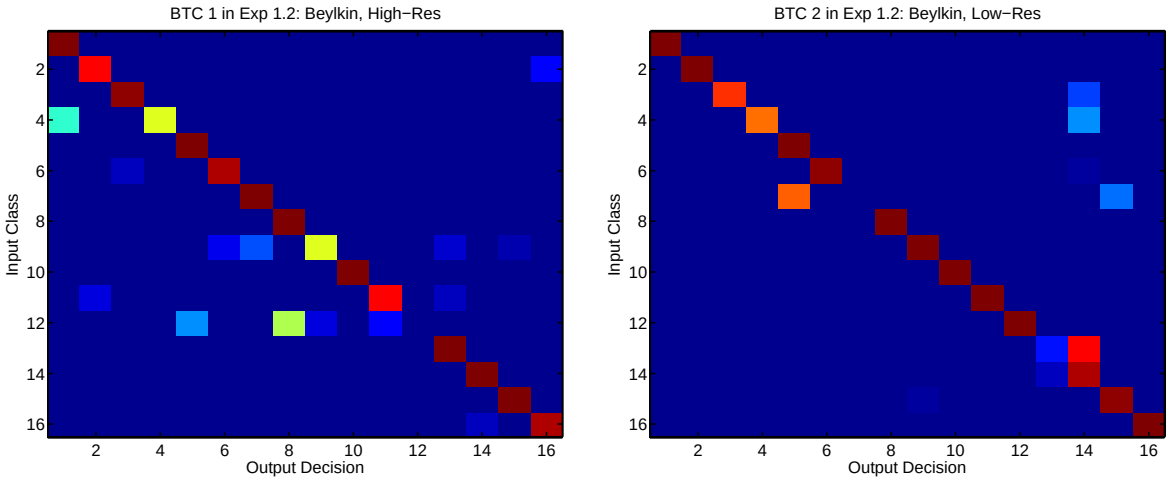


Figure 25: Confusion matrices for BTCs 1 and 2 in Experiment 1.2.

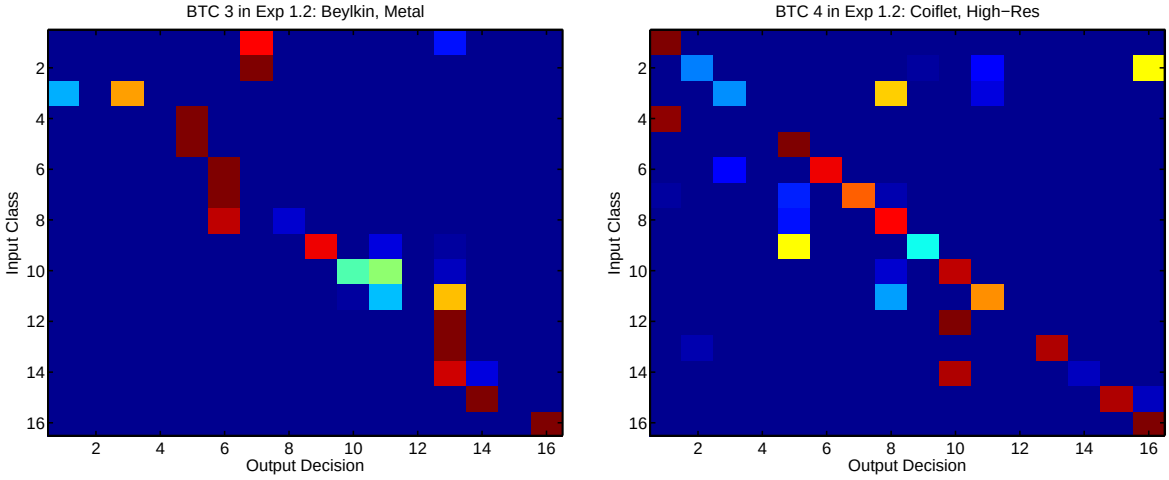


Figure 26: Confusion matrices for BTCs 3 and 4 in Experiment 1.2.

have ambiguities near zero, but are also highly susceptible to noise, so that the hypertree algorithm must take both metrics into account when choosing nodes. The current version of the algorithm does, in fact, take both of these parameters into account, but the method must be refined.

3. The average quality measure for incorrect decisions is very small compared to the average quality measure for correct decisions. This implies that path correction may provide a substantial performance boost for all BTCs and BSCs.

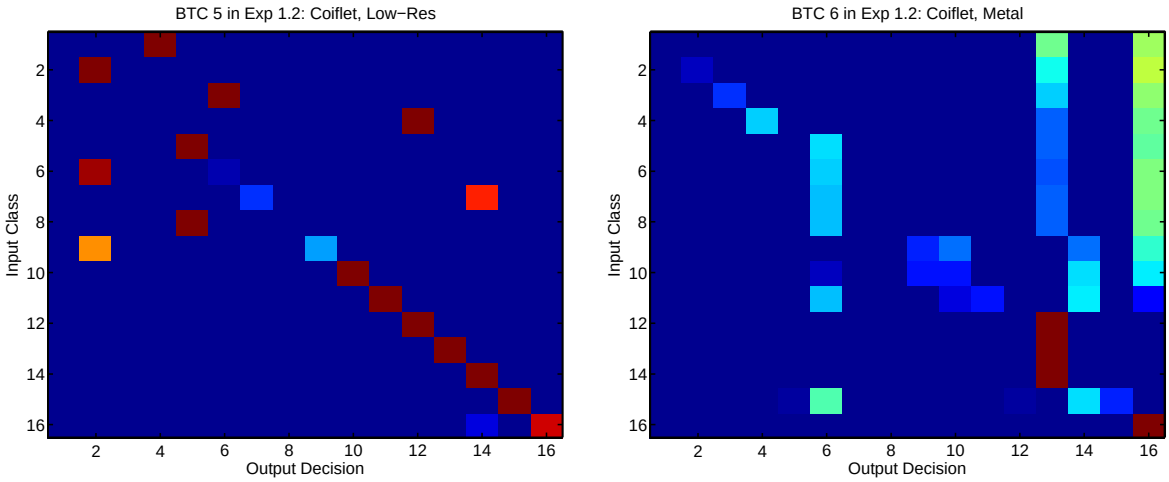


Figure 27: Confusion matrices for BTCs 5 and 6 in Experiment 1.2.

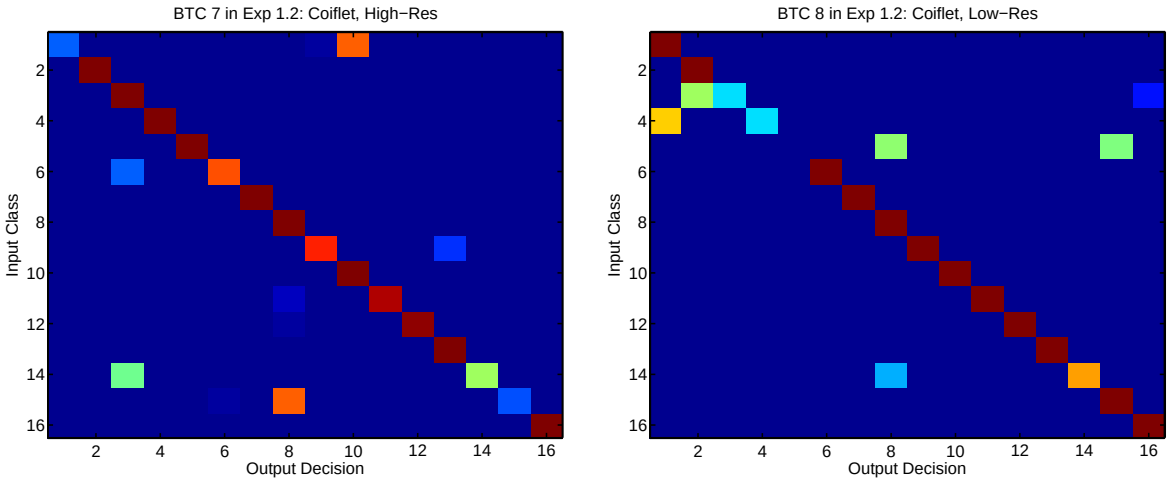


Figure 28: Confusion matrices for BTCs 7 and 8 in Experiment 1.2.

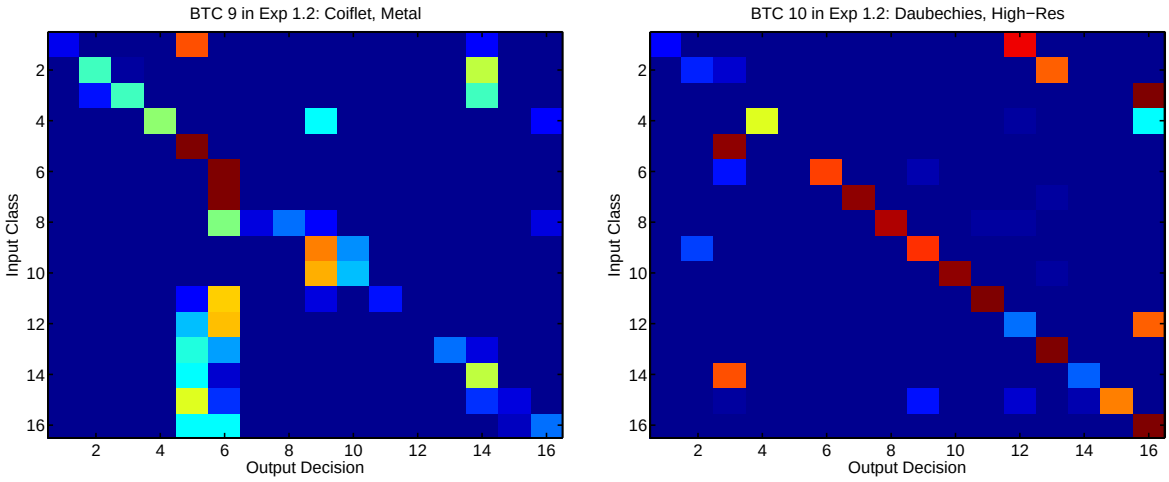


Figure 29: Confusion matrices for BTCs 9 and 10 in Experiment 1.2.

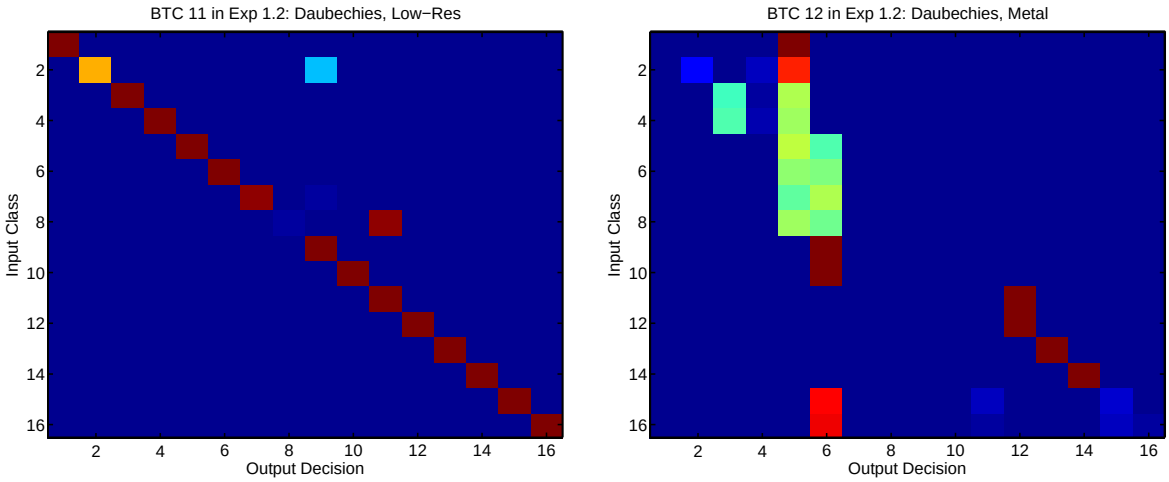


Figure 30: Confusion matrices for BTCs 11 and 12 in Experiment 1.2.

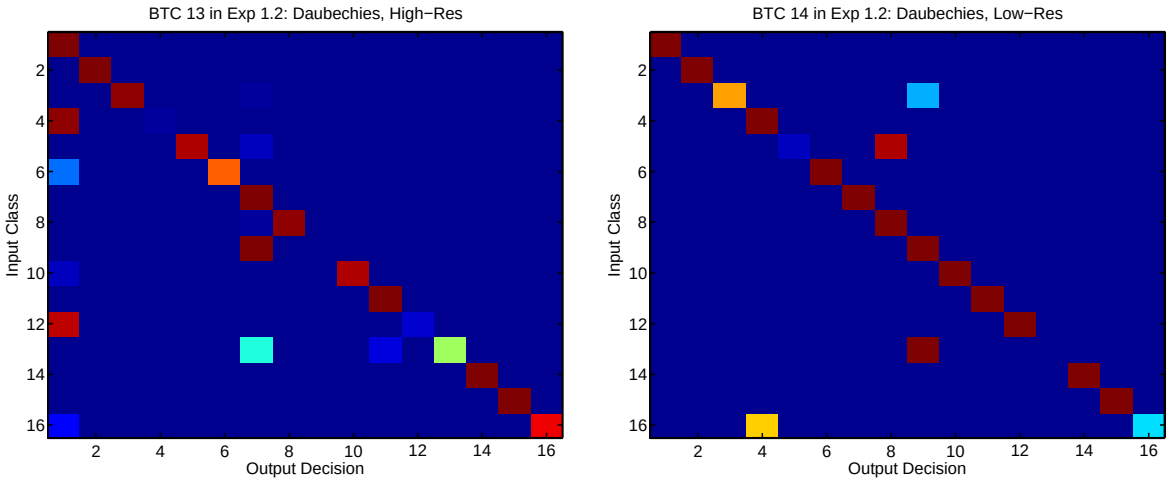


Figure 31: Confusion matrices for BTCs 13 and 14 in Experiment 1.2.

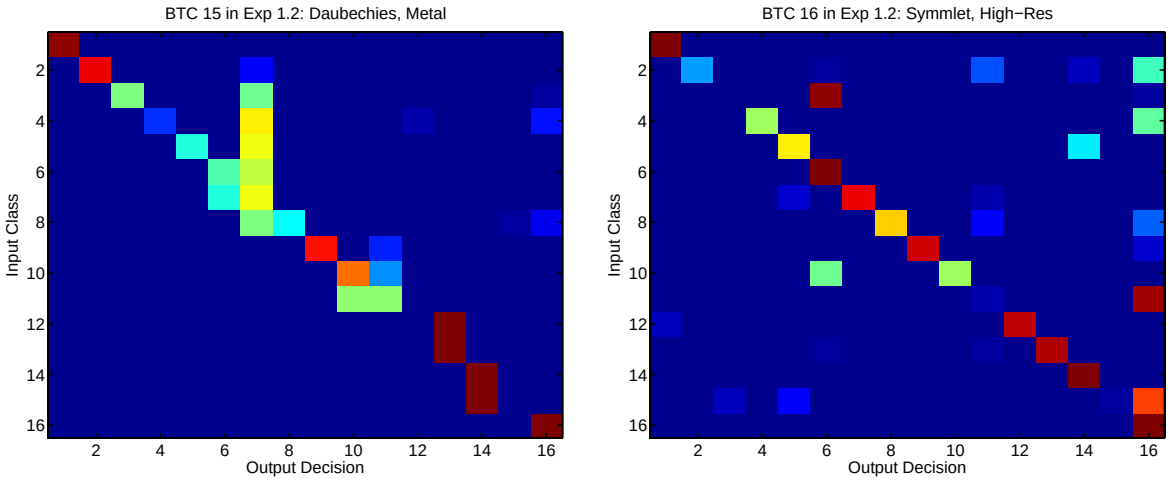


Figure 32: Confusion matrices for BTCs 15 and 16 in Experiment 1.2.

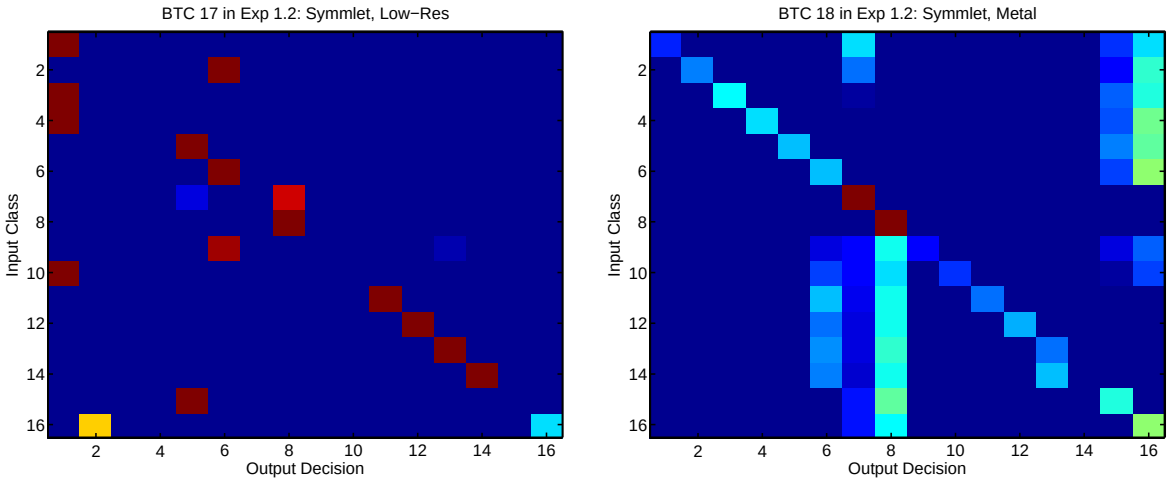


Figure 33: Confusion matrices for BTCs 17 and 18 in Experiment 1.2.

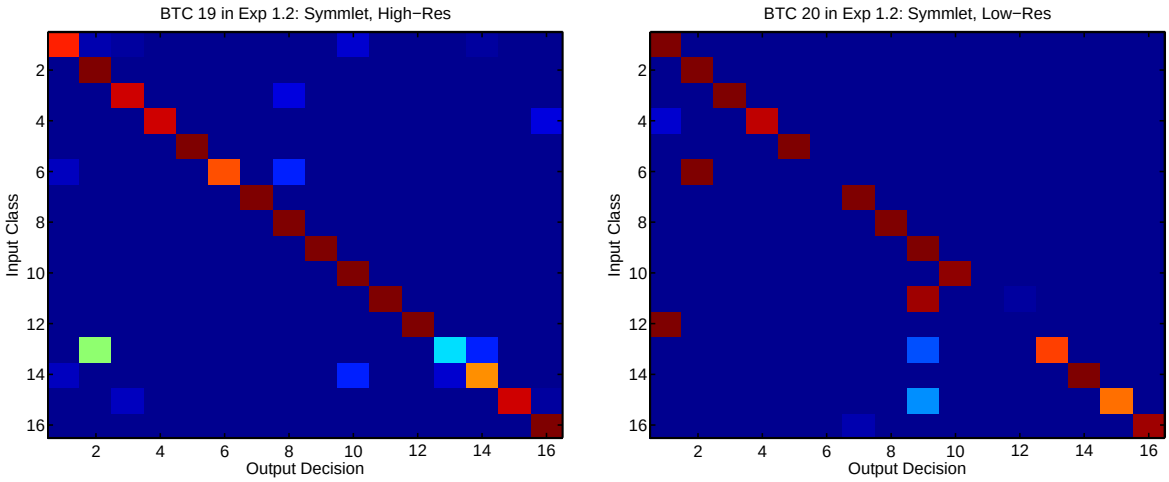


Figure 34: Confusion matrices for BTCs 19 and 20 in Experiment 1.2.

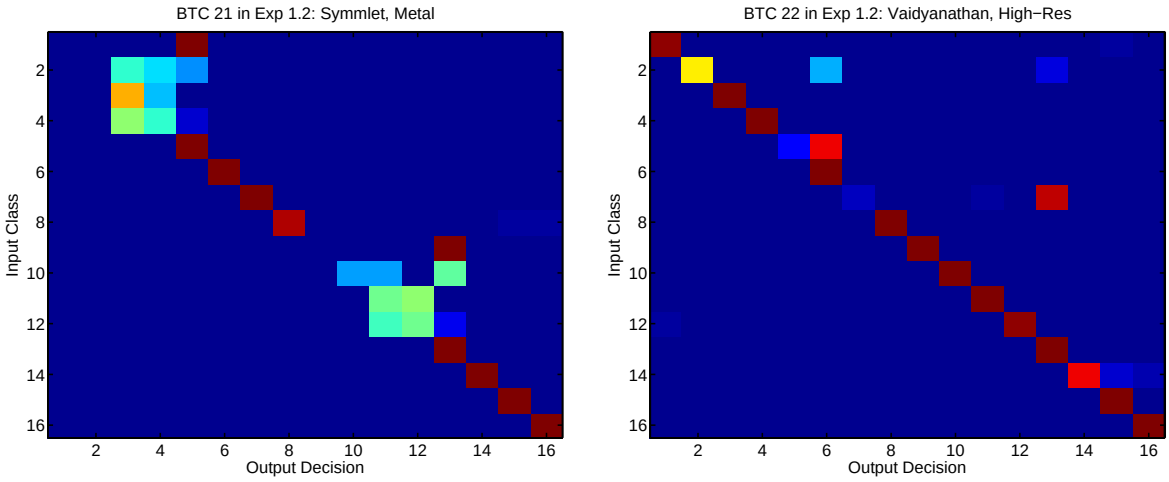


Figure 35: Confusion matrices for BTCs 21 and 22 in Experiment 1.2.

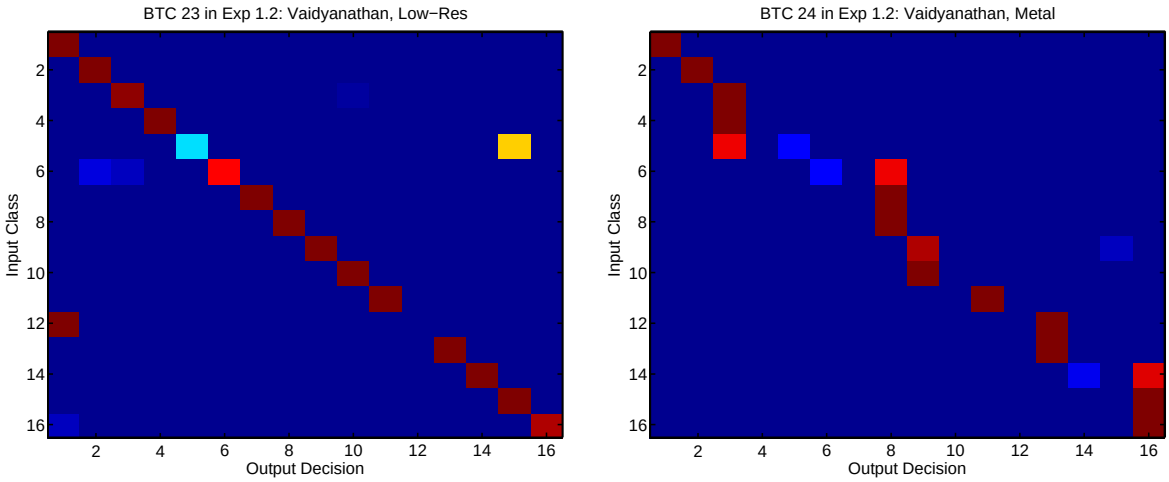


Figure 36: Confusion matrices for BTCs 23 and 24 in Experiment 1.2.

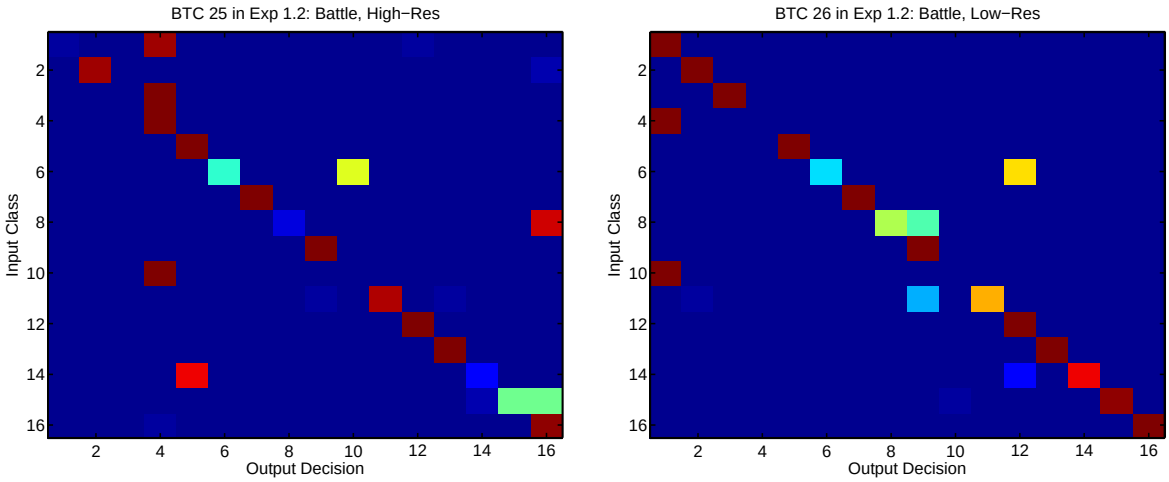


Figure 37: Confusion matrices for BTCs 25 and 26 in Experiment 1.2.

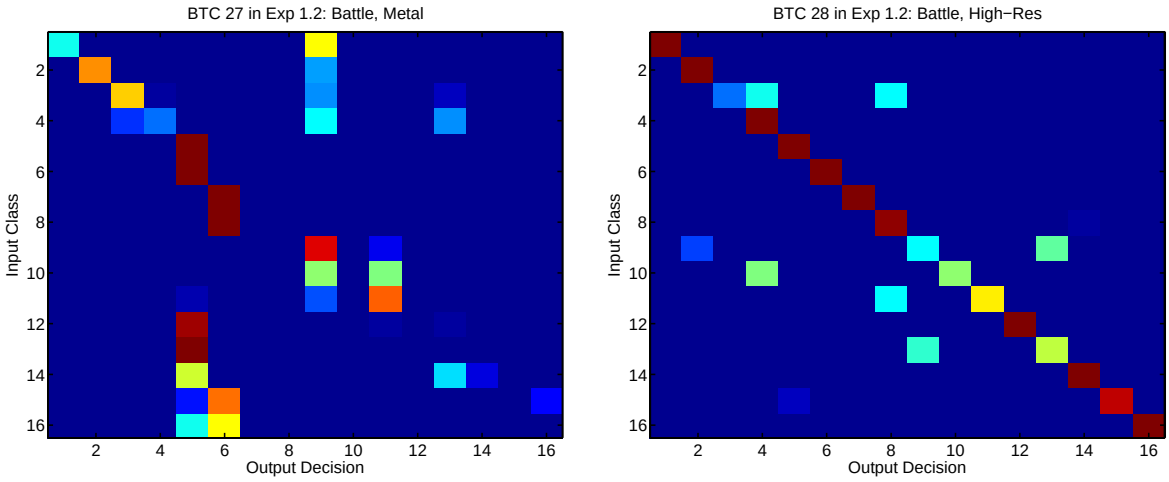


Figure 38: Confusion matrices for BTCs 27 and 28 in Experiment 1.2.

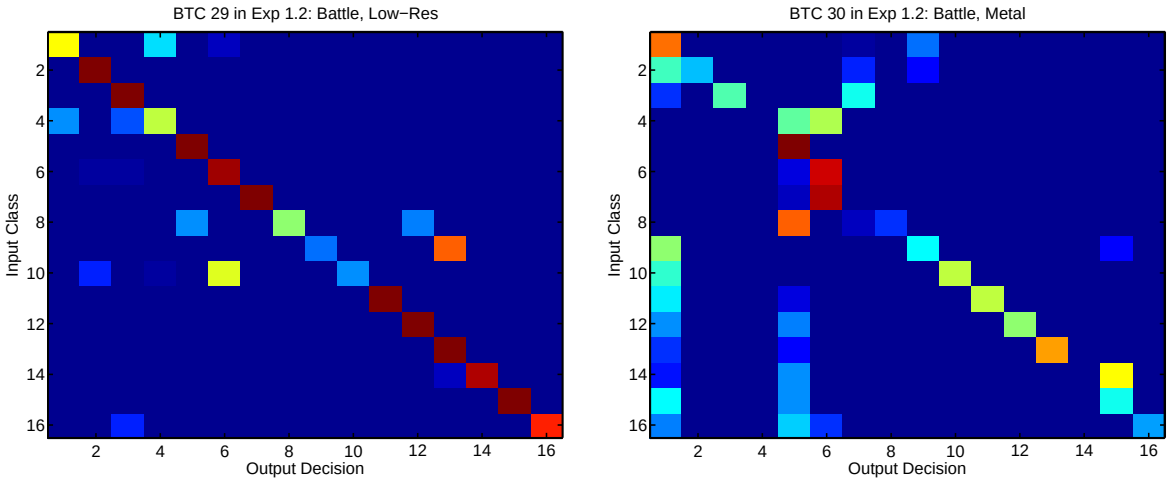


Figure 39: Confusion matrices for BTCs 29 and 30 in Experiment 1.2.

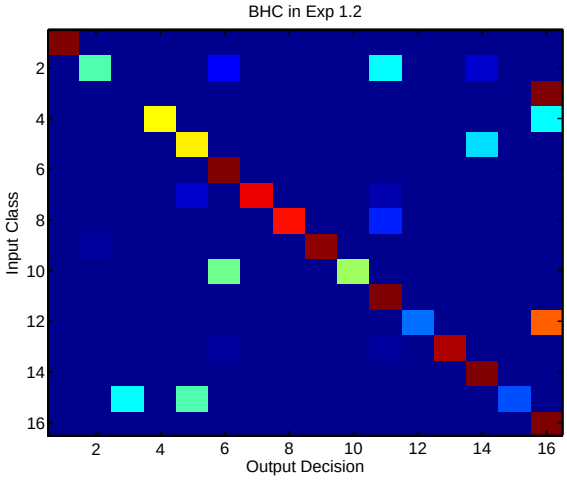


Figure 40: Confusion matrix for the BHC in Experiment 1.2.

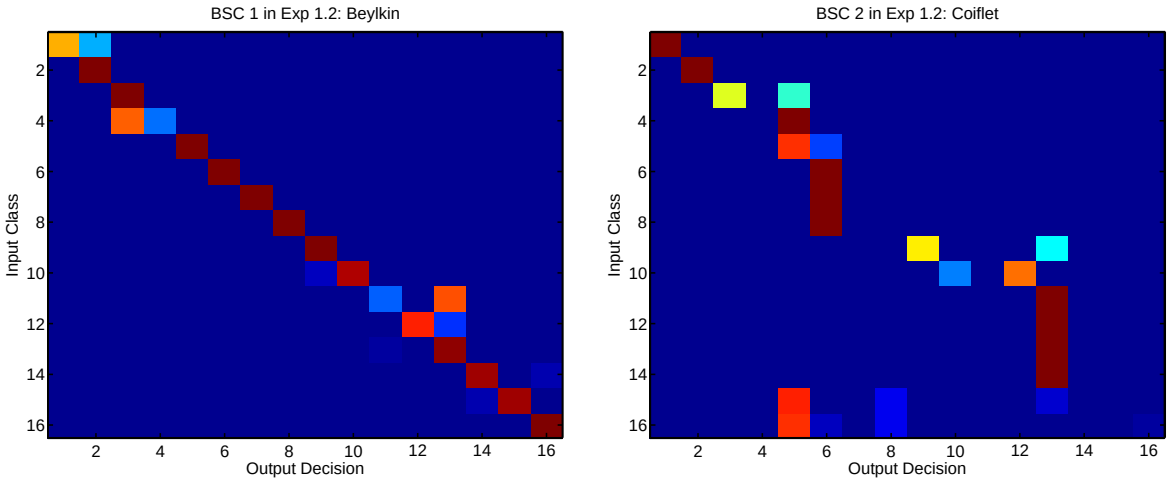


Figure 41: Confusion matrices for BSCs 1 and 2 in Experiment 1.2.

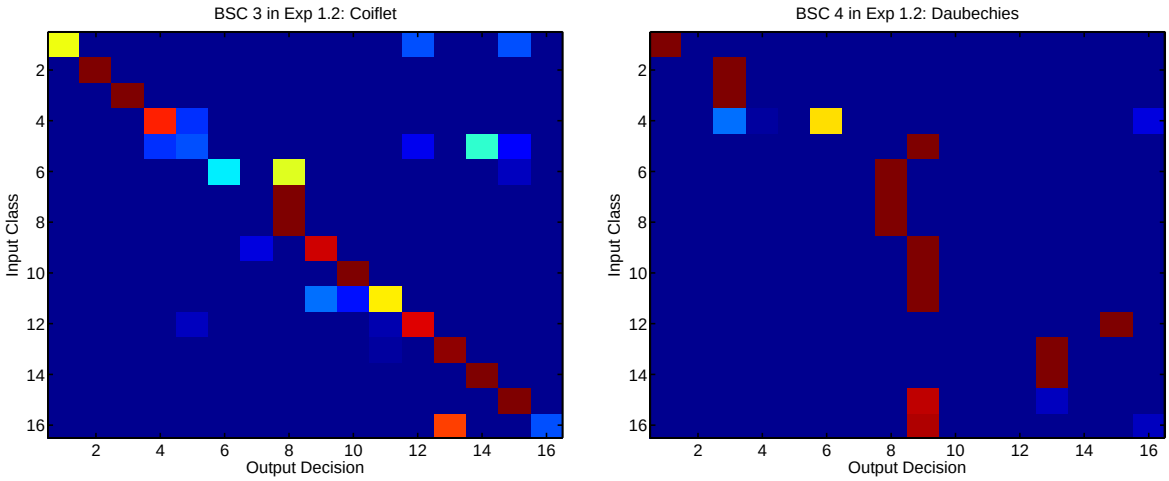


Figure 42: Confusion matrices for BSCs 3 and 4 in Experiment 1.2.

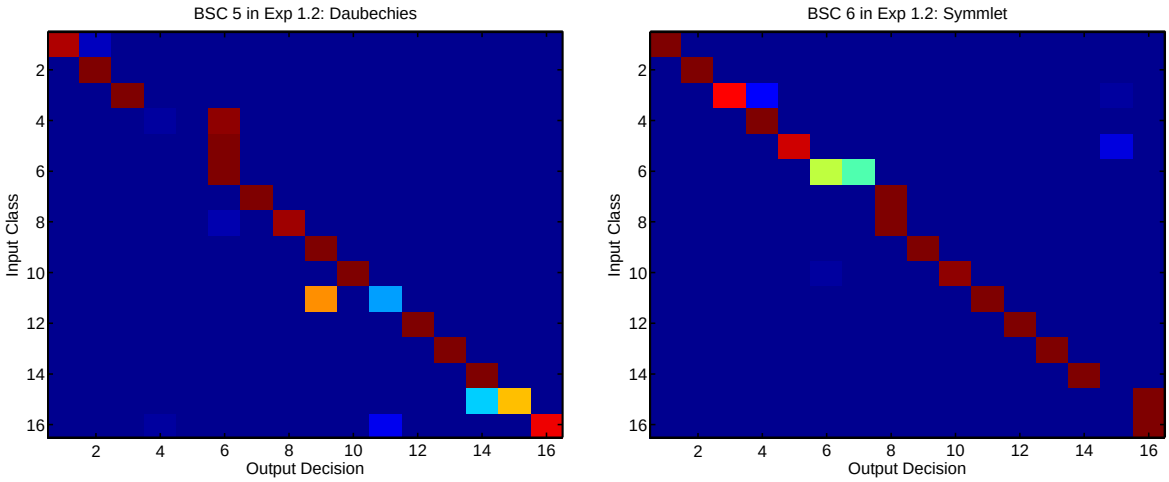


Figure 43: Confusion matrices for BSCs 5 and 6 in Experiment 1.2.

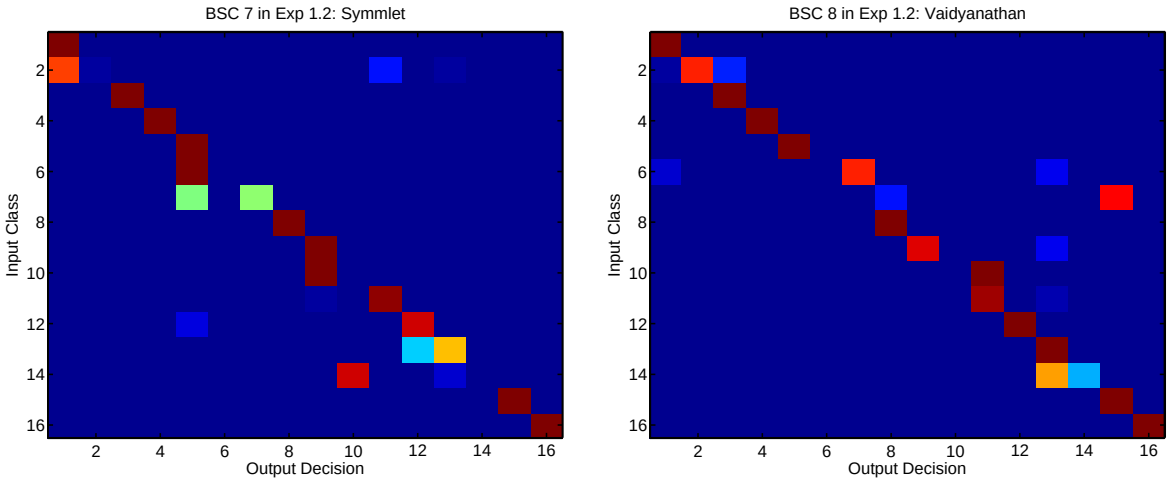


Figure 44: Confusion matrices for BSCs 7 and 8 in Experiment 1.2.

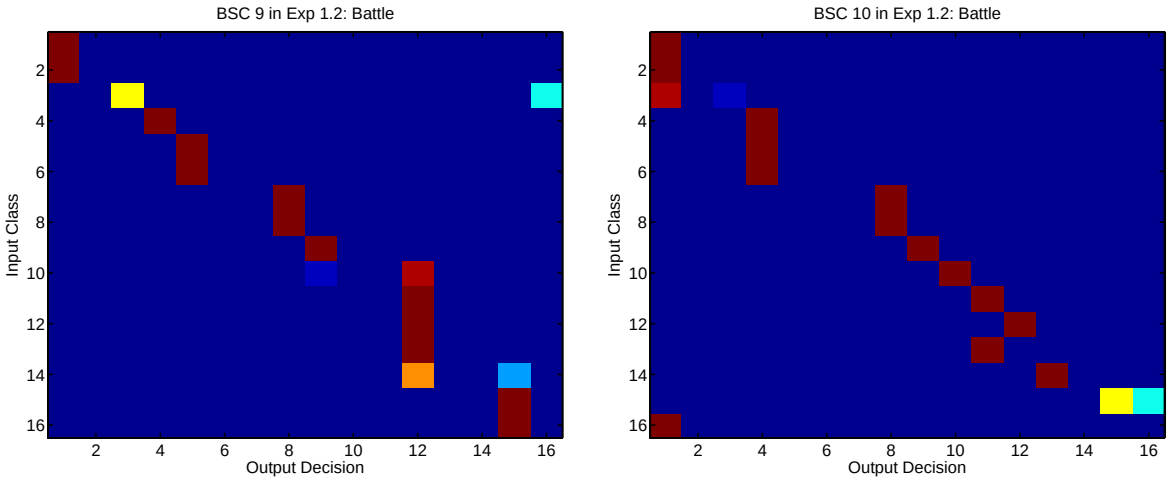


Figure 45: Confusion matrices for BSCs 9 and 10 in Experiment 1.2.

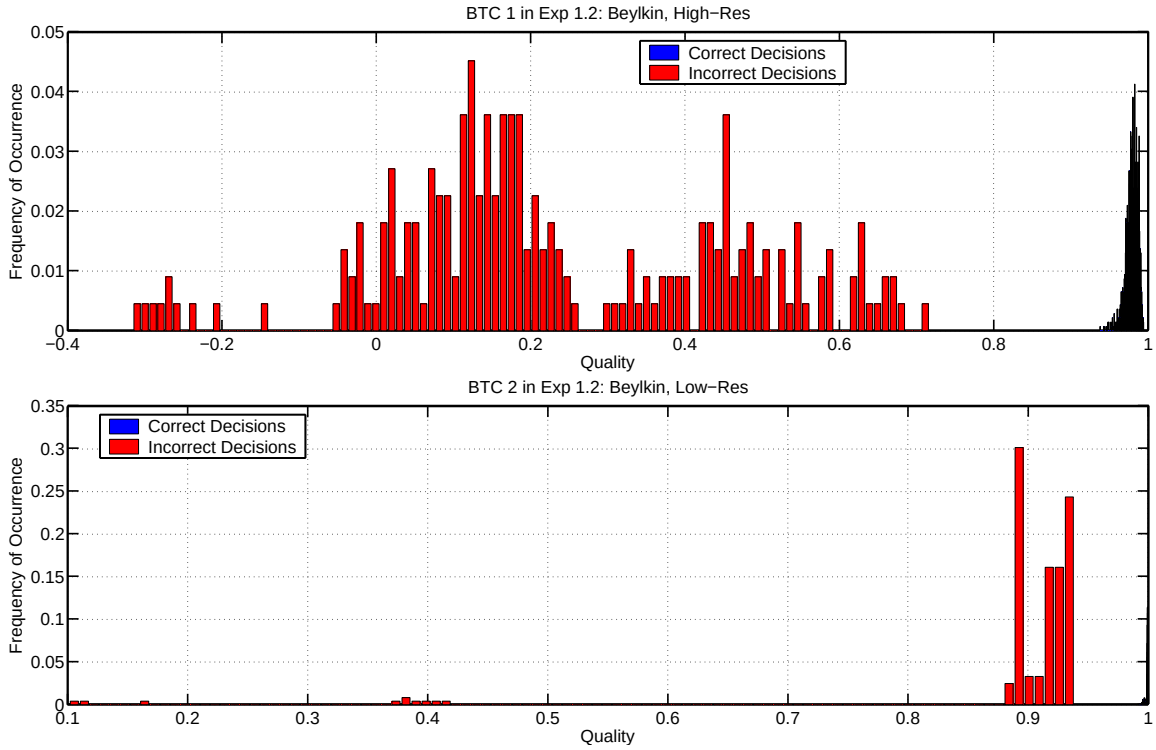


Figure 46: Quality histograms for BTCs 1 and 2 in Experiment 1.2.

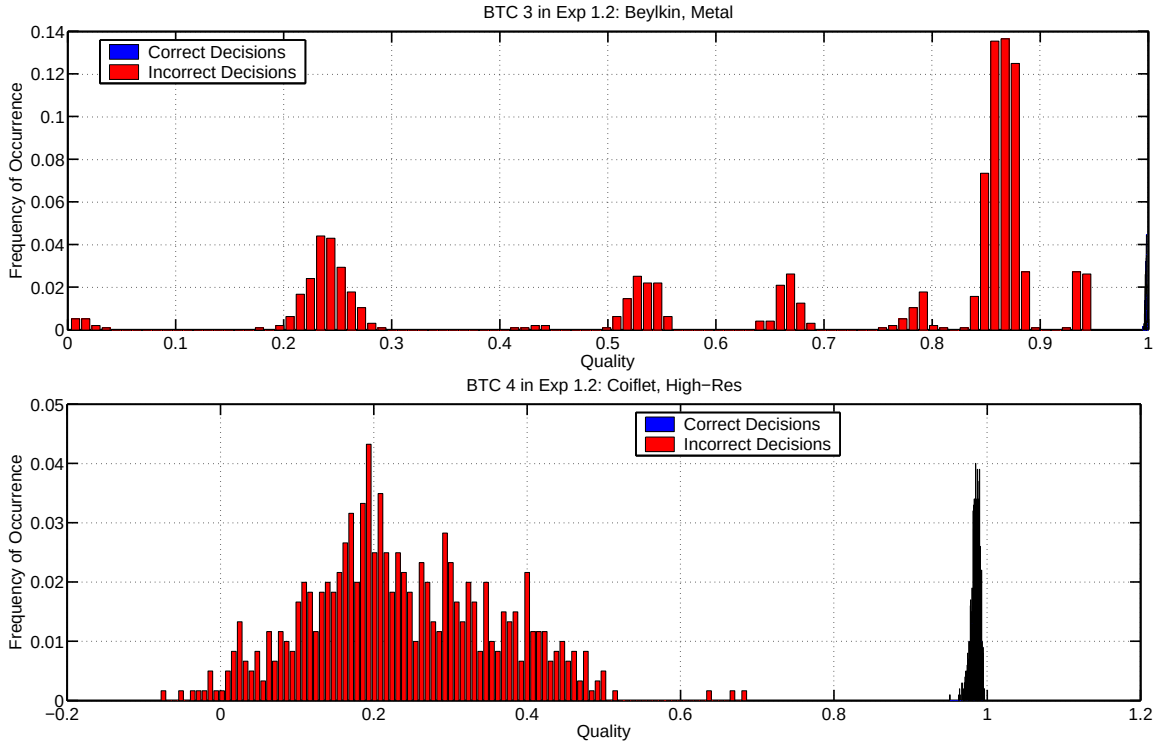


Figure 47: Quality histograms for BTCs 3 and 4 in Experiment 1.2.

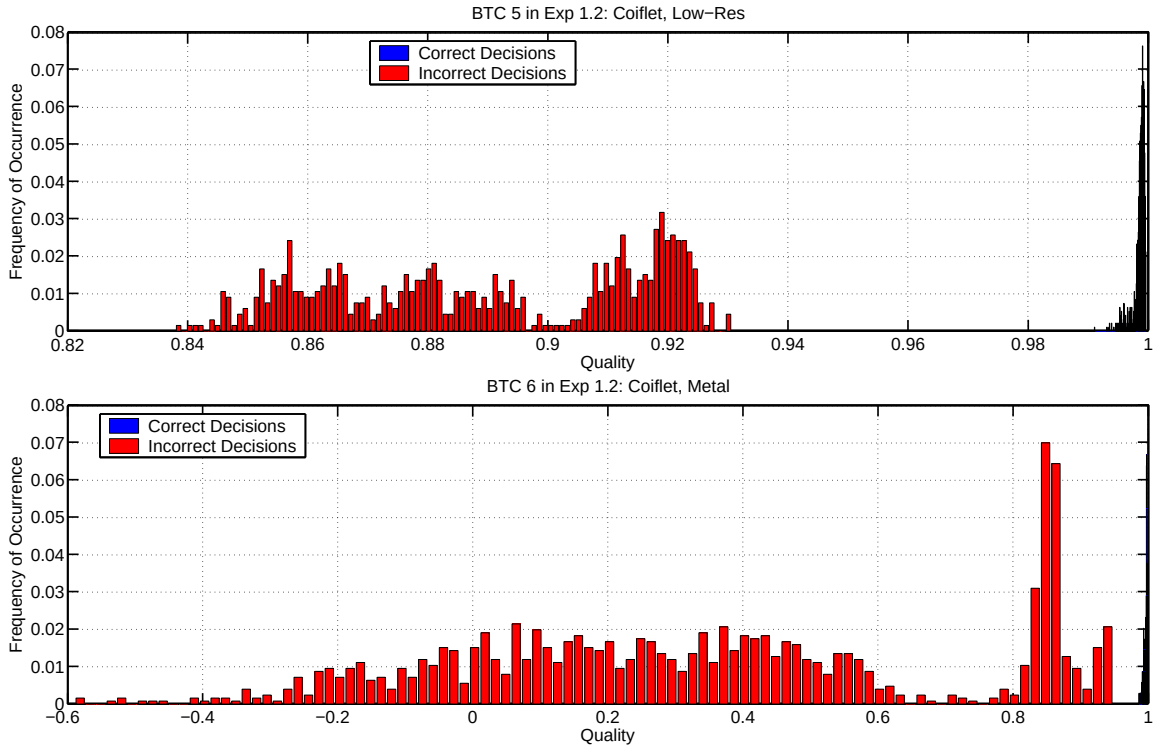


Figure 48: Quality histograms for BTCs 5 and 6 in Experiment 1.2.

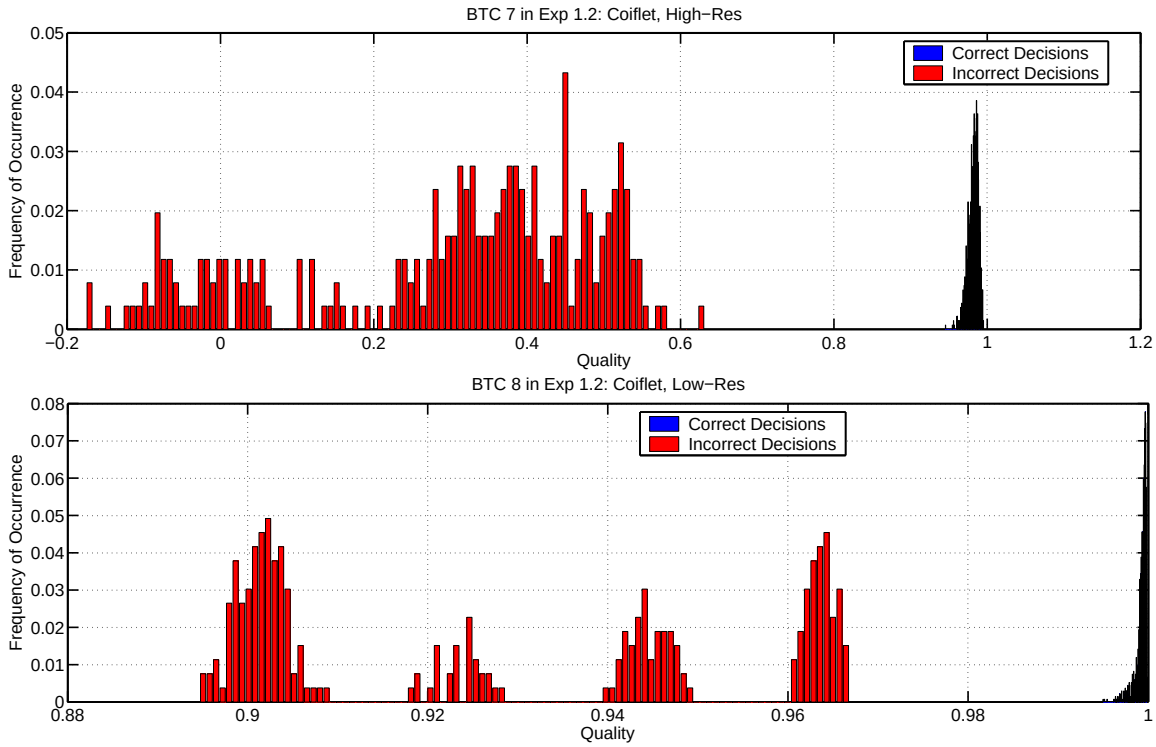


Figure 49: Quality histograms for BTCs 7 and 8 in Experiment 1.2.

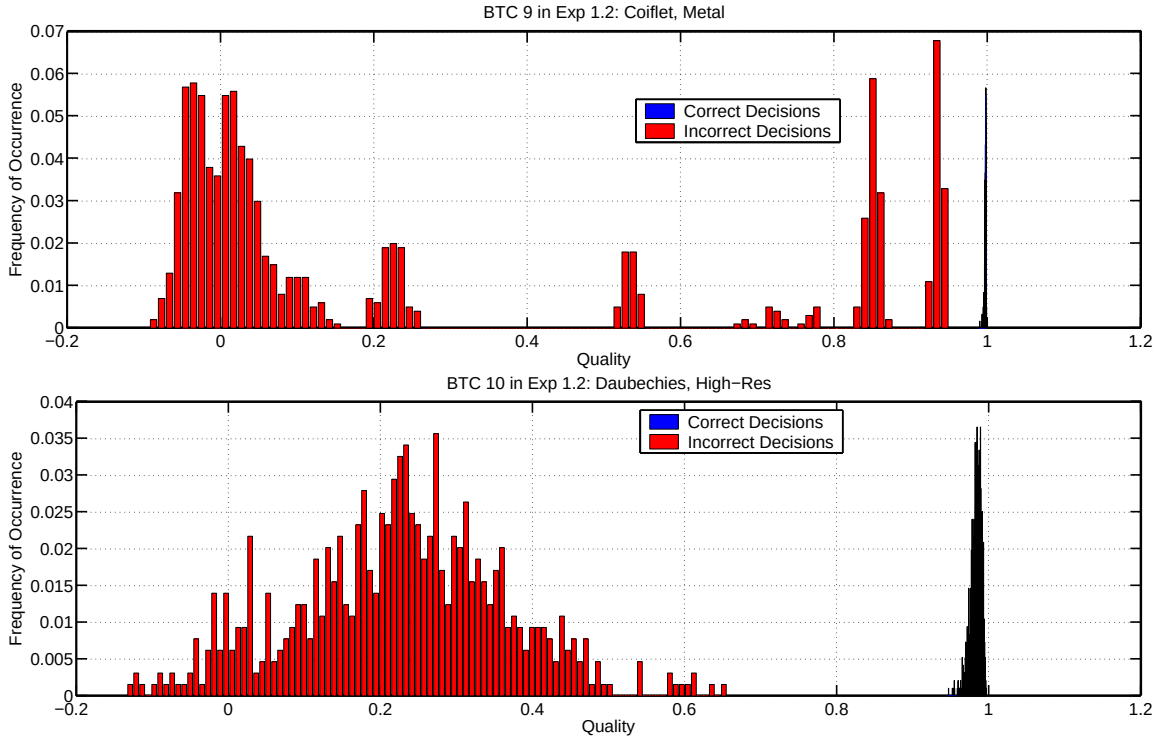


Figure 50: Quality histograms for BTCs 9 and 10 in Experiment 1.2.

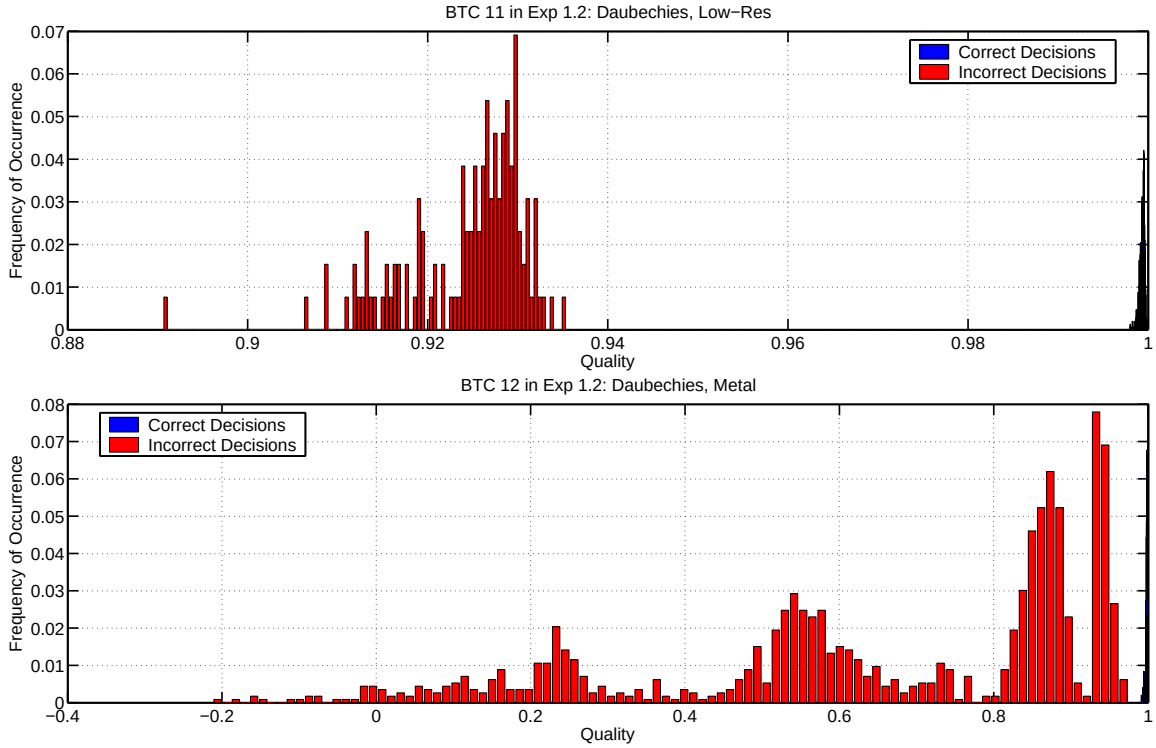


Figure 51: Quality histograms for BTCs 11 and 12 in Experiment 1.2.

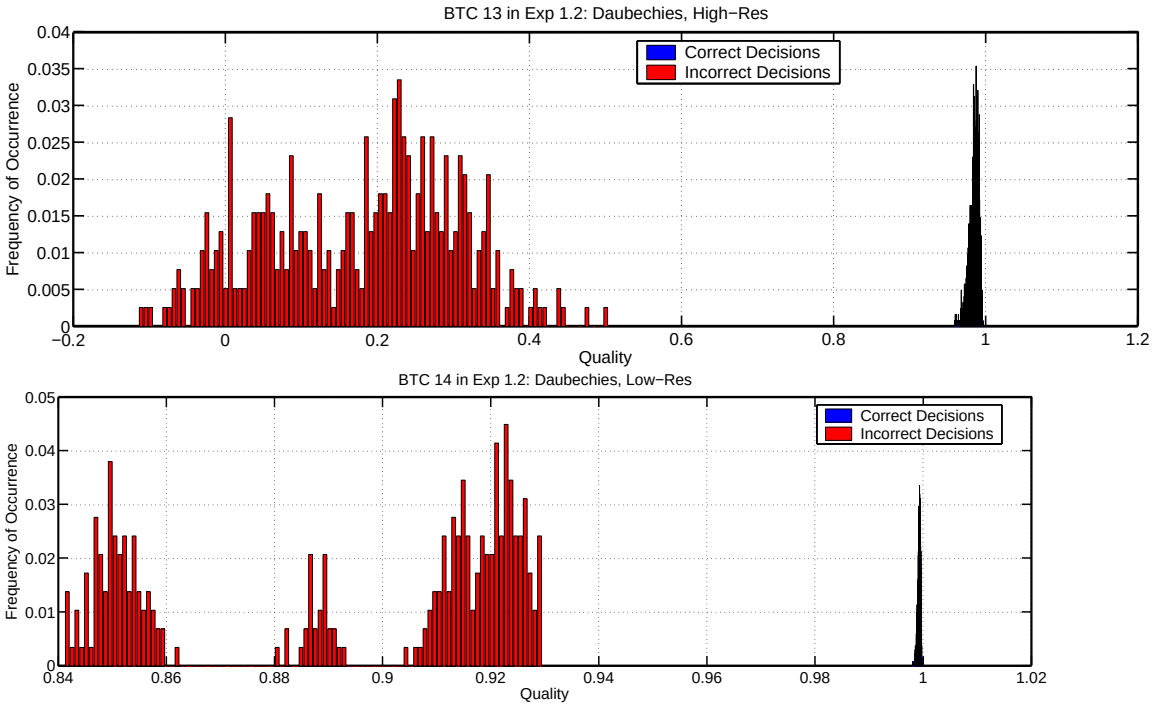


Figure 52: Quality histograms for BTCs 13 and 14 in Experiment 1.2.

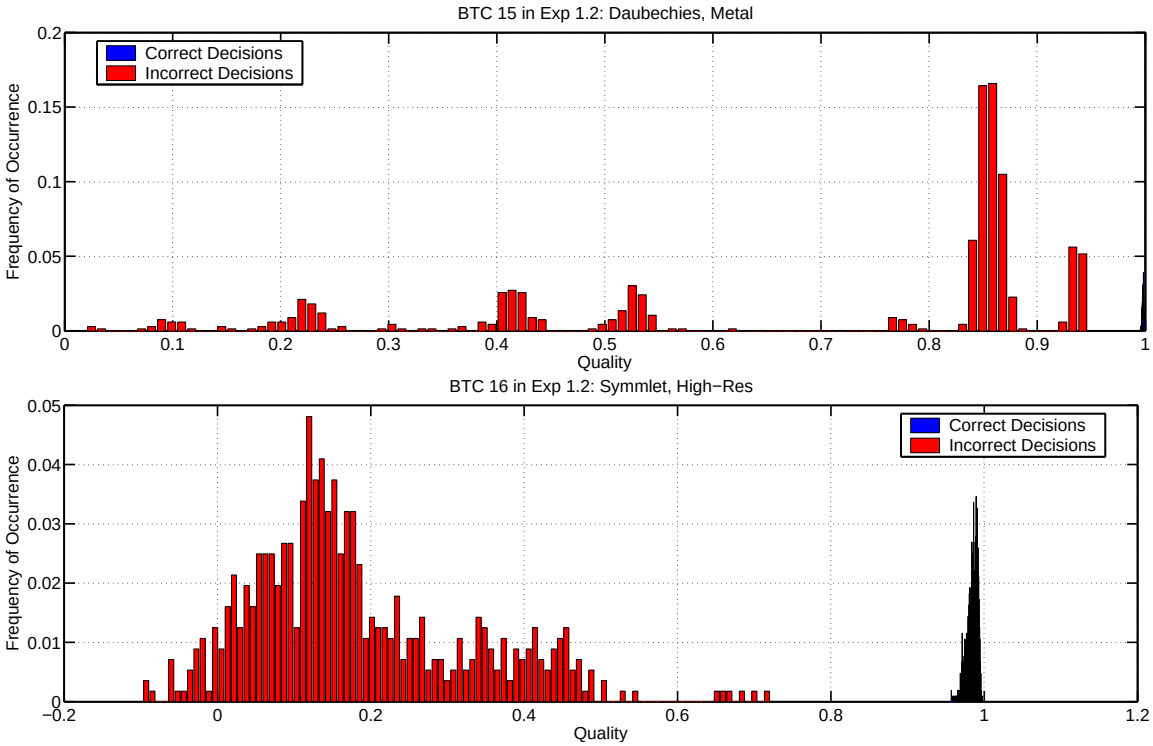


Figure 53: Quality histograms for BTCs 15 and 16 in Experiment 1.2.

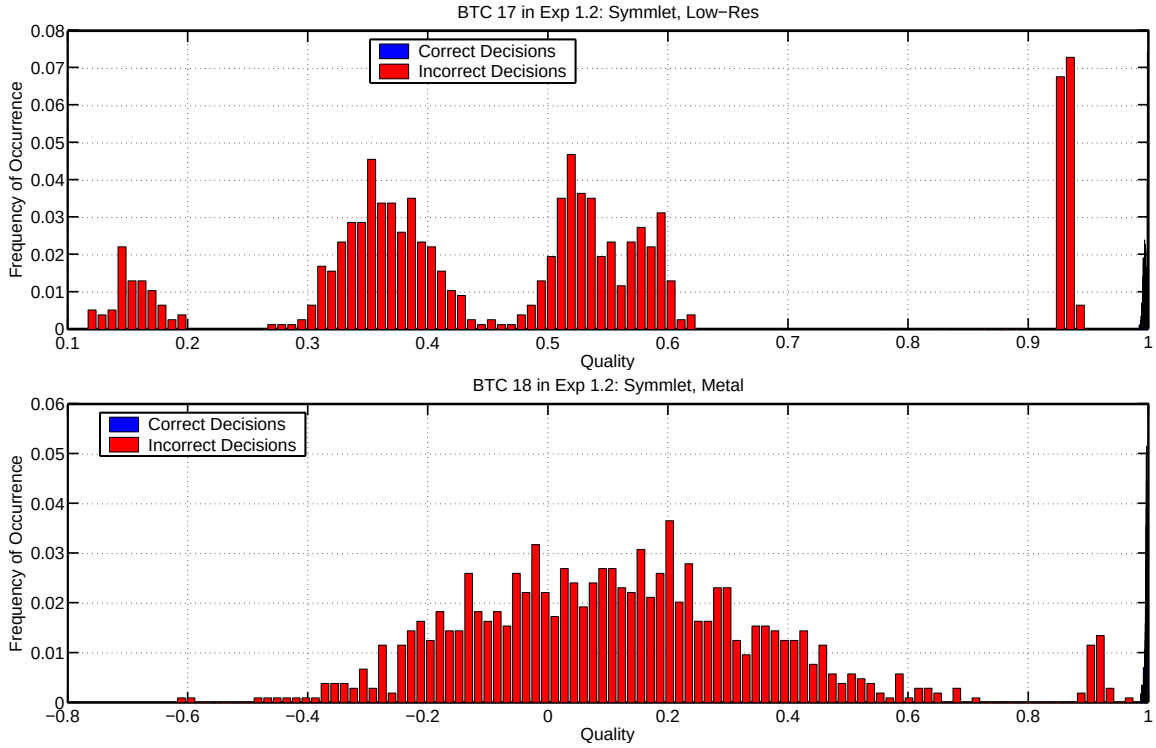


Figure 54: Quality histograms for BTCs 17 and 18 in Experiment 1.2.

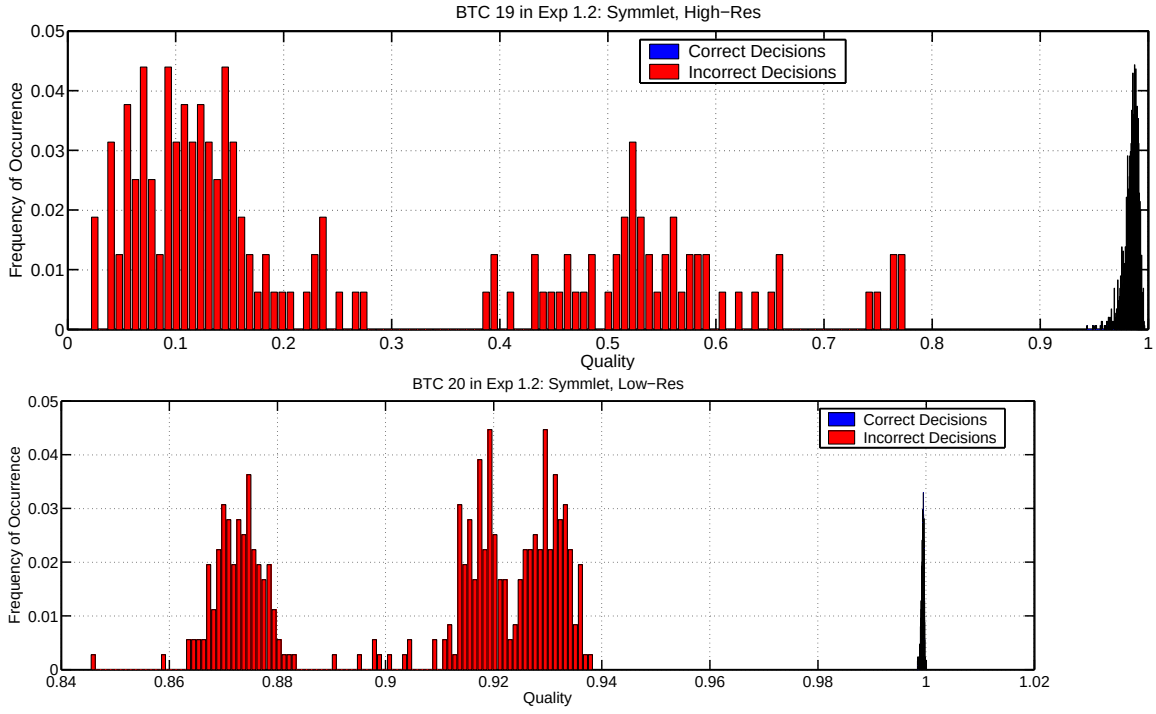


Figure 55: Quality histograms for BTCs 19 and 20 in Experiment 1.2.

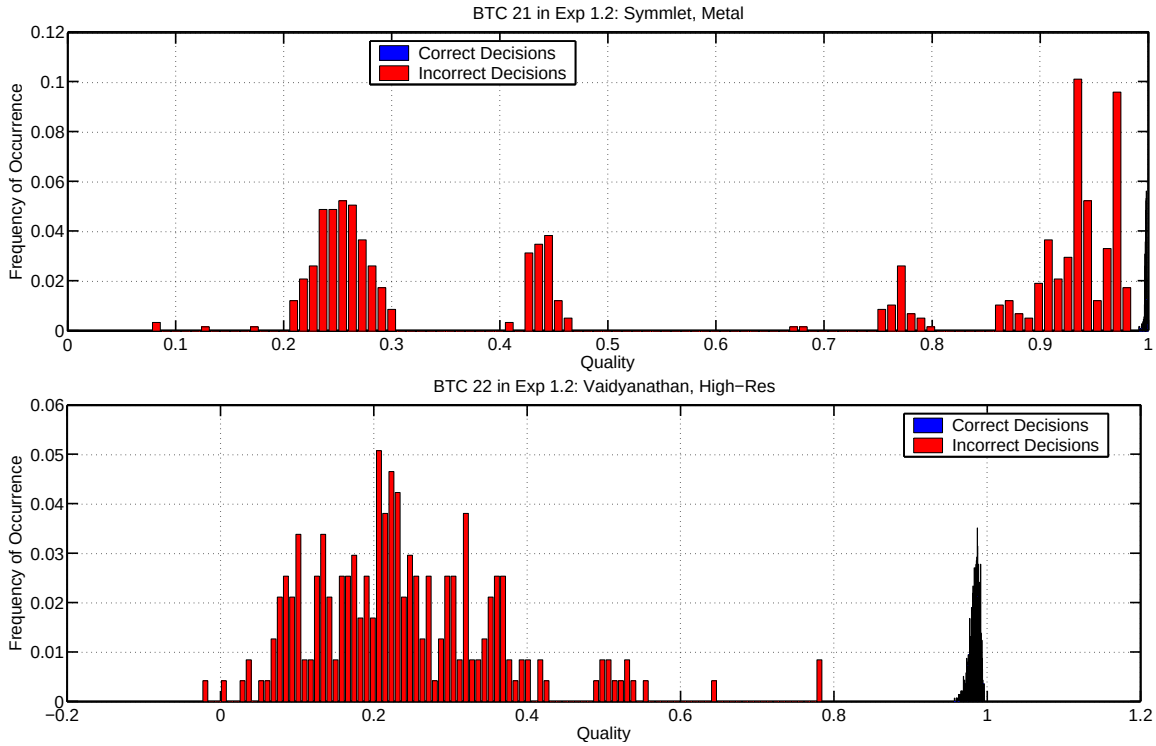


Figure 56: Quality histograms for BTCs 21 and 22 in Experiment 1.2.

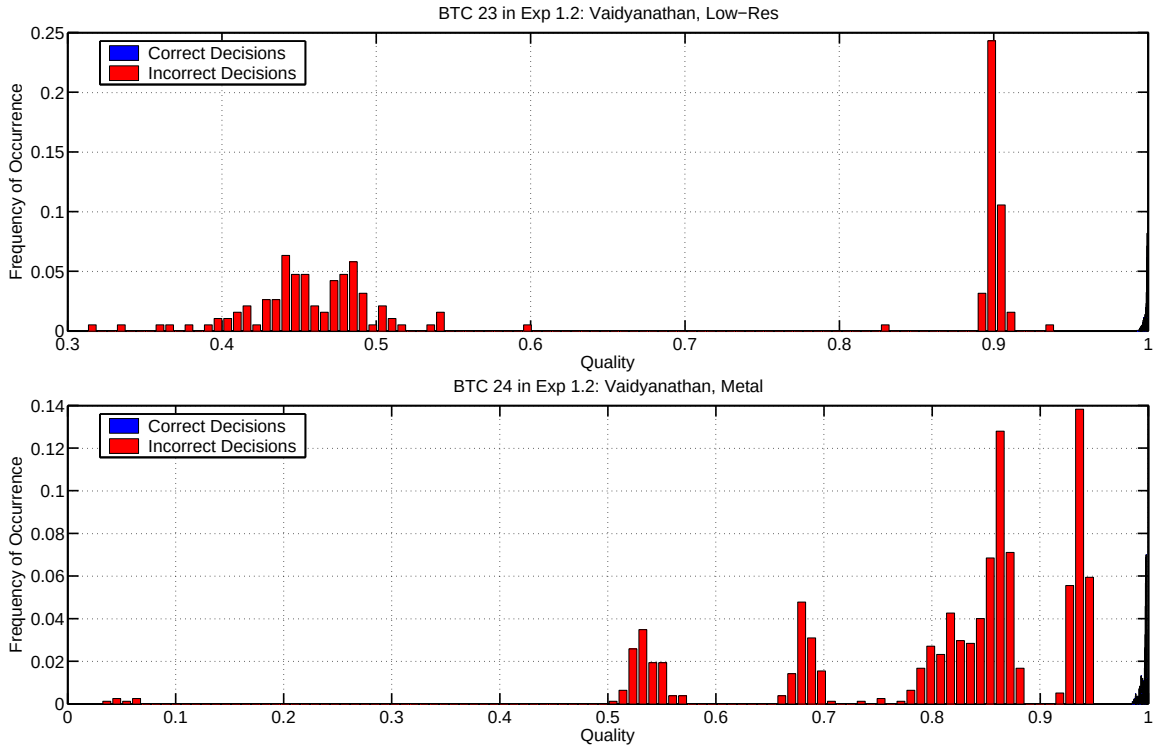


Figure 57: Quality histograms for BTCs 23 and 24 in Experiment 1.2.

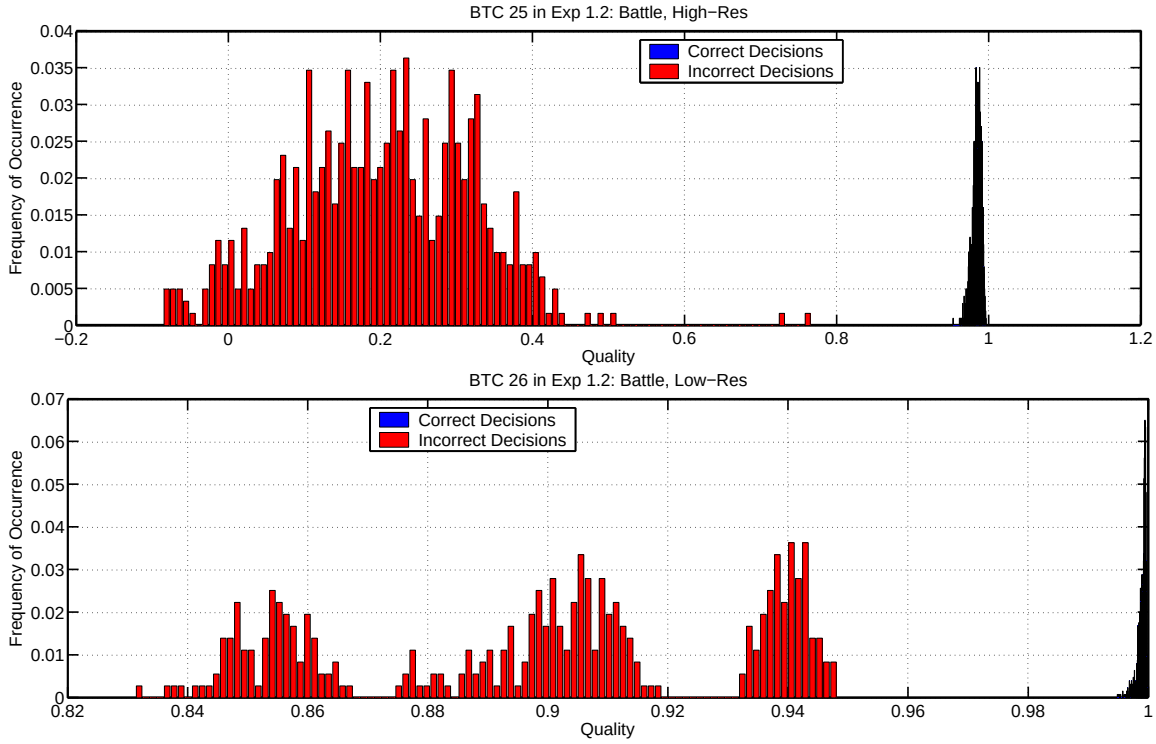


Figure 58: Quality histograms for BTCs 25 and 26 in Experiment 1.2.

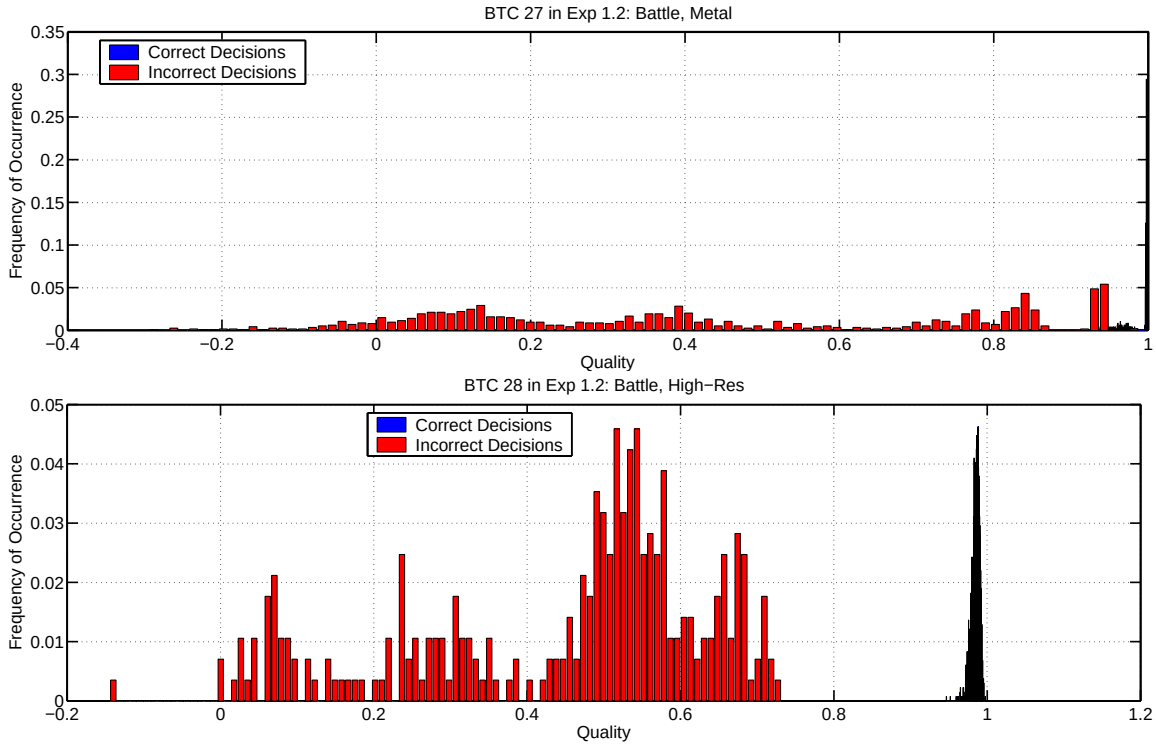


Figure 59: Quality histograms for BTCs 27 and 28 in Experiment 1.2.

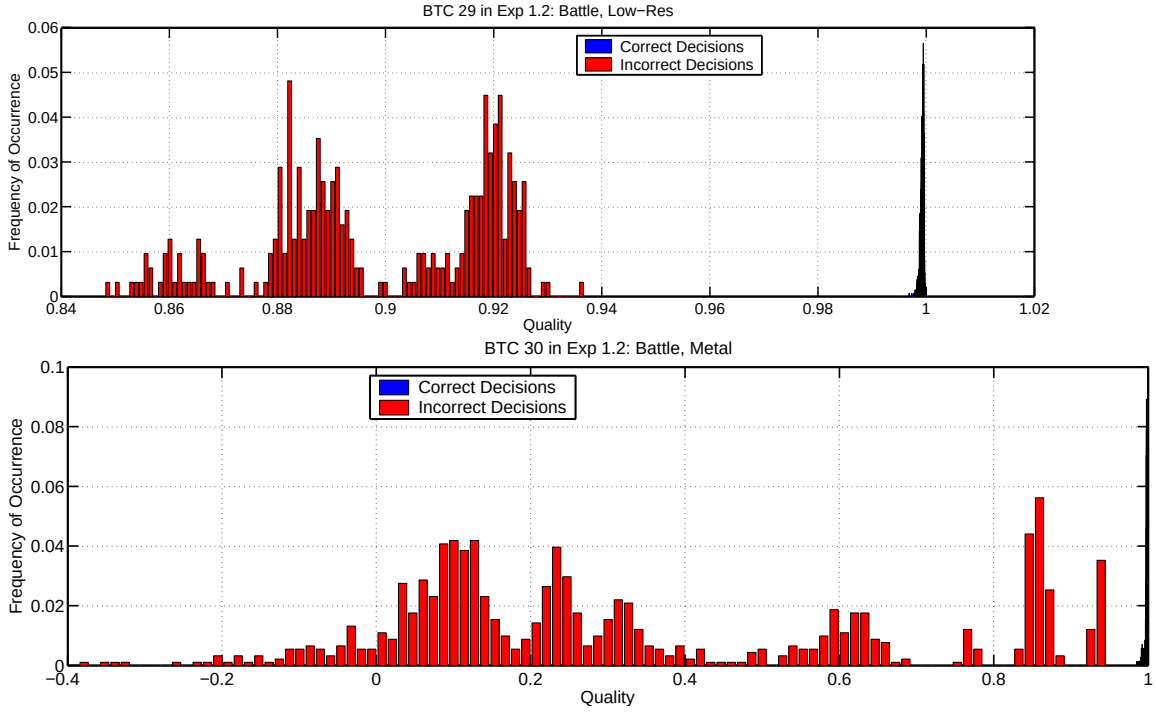


Figure 60: Quality histograms for BTCs 29 and 30 in Experiment 1.2.

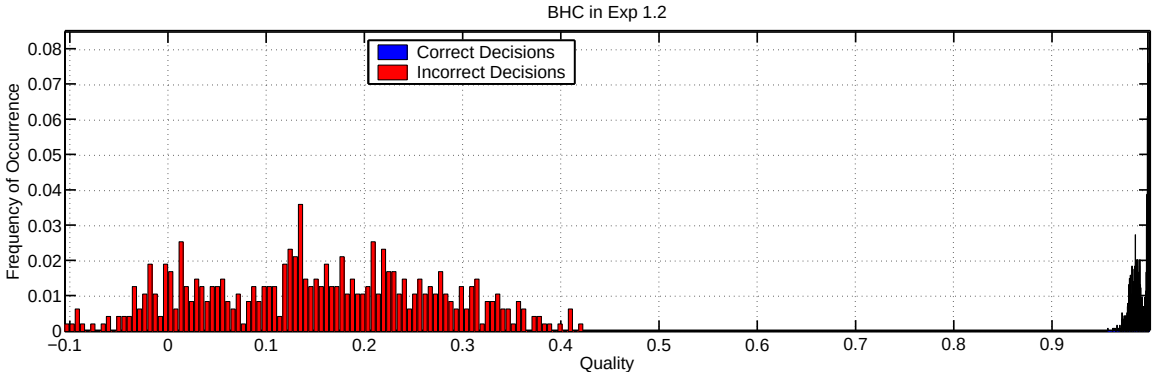


Figure 61: Quality histograms for the BHC in Experiment 1.2.

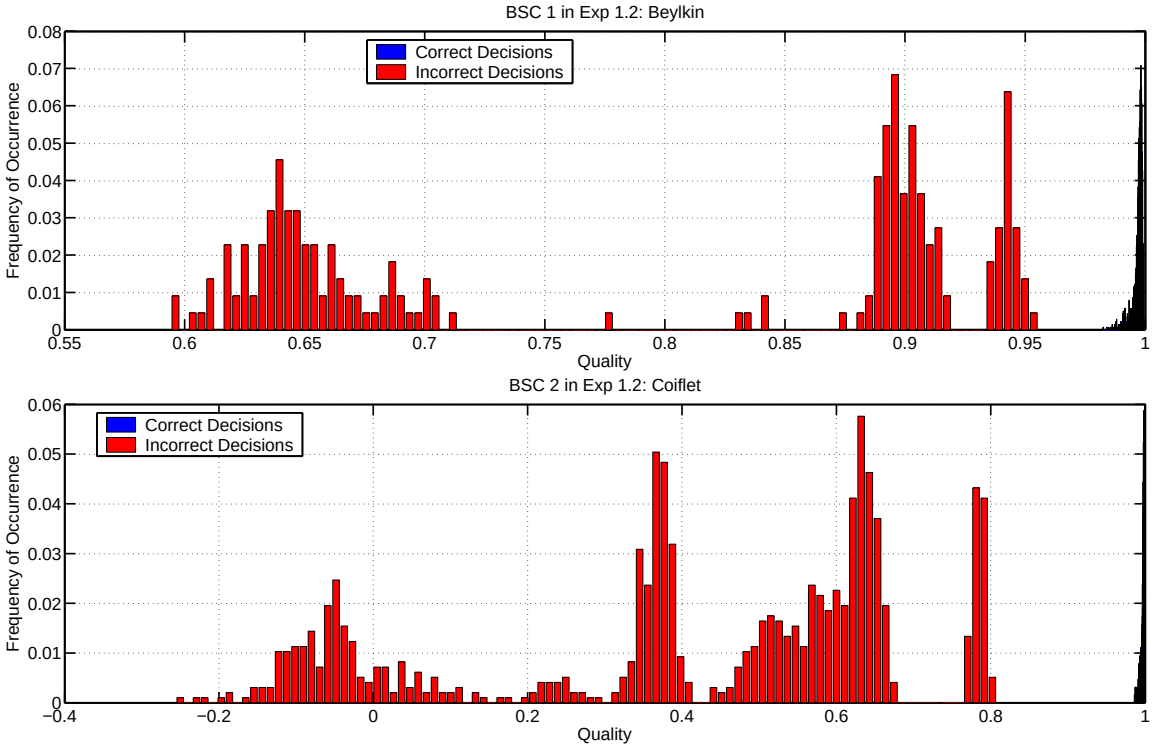


Figure 62: Quality histograms for BSCs 1 and 2 in Experiment 1.2.

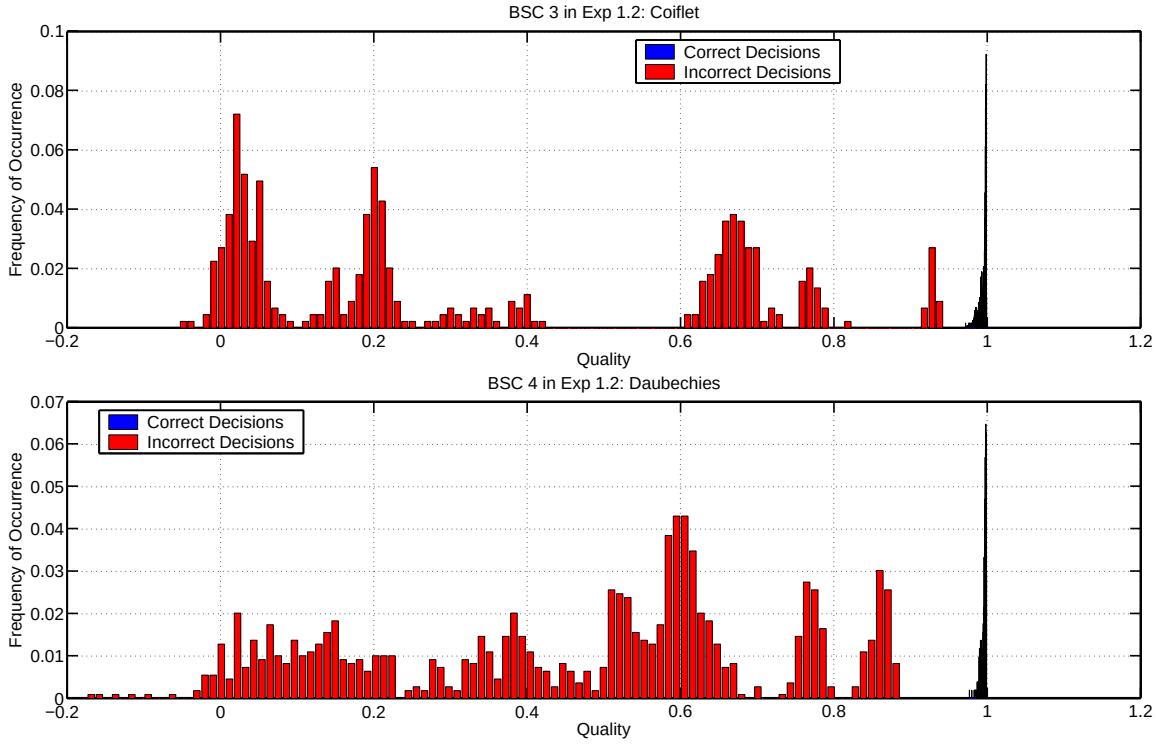


Figure 63: Quality histograms for BSCs 3 and 4 in Experiment 1.2.

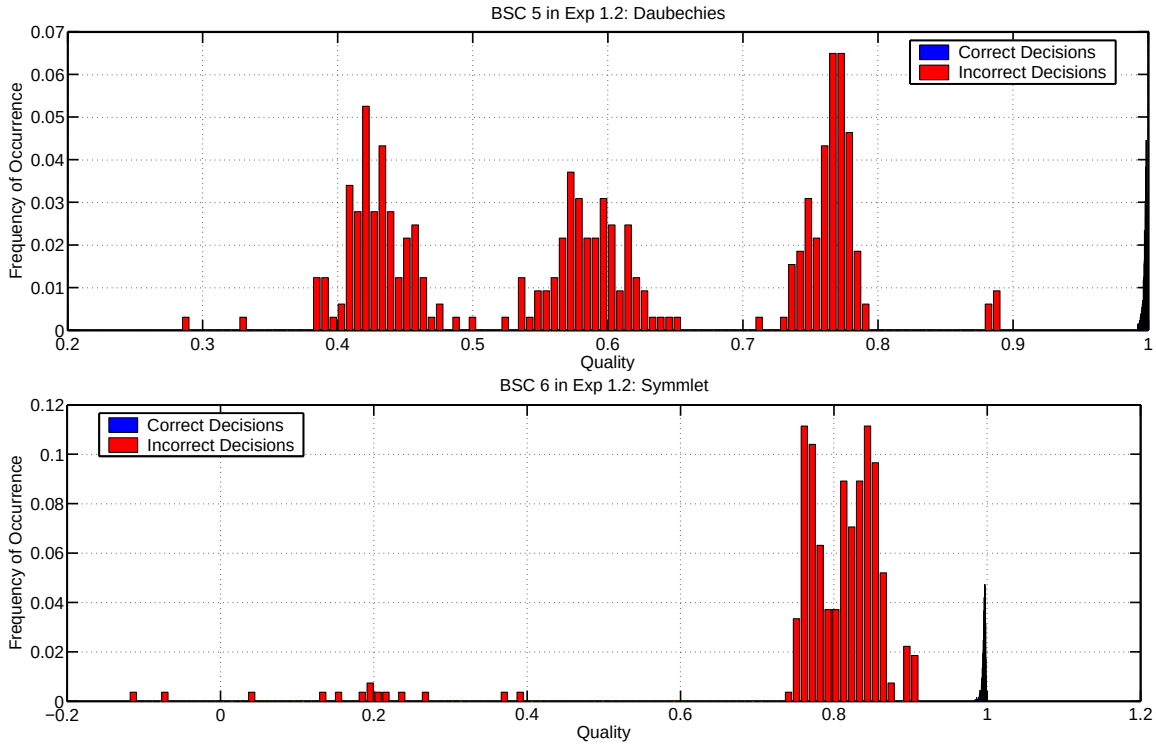


Figure 64: Quality histograms for BSCs 5 and 6 in Experiment 1.2.

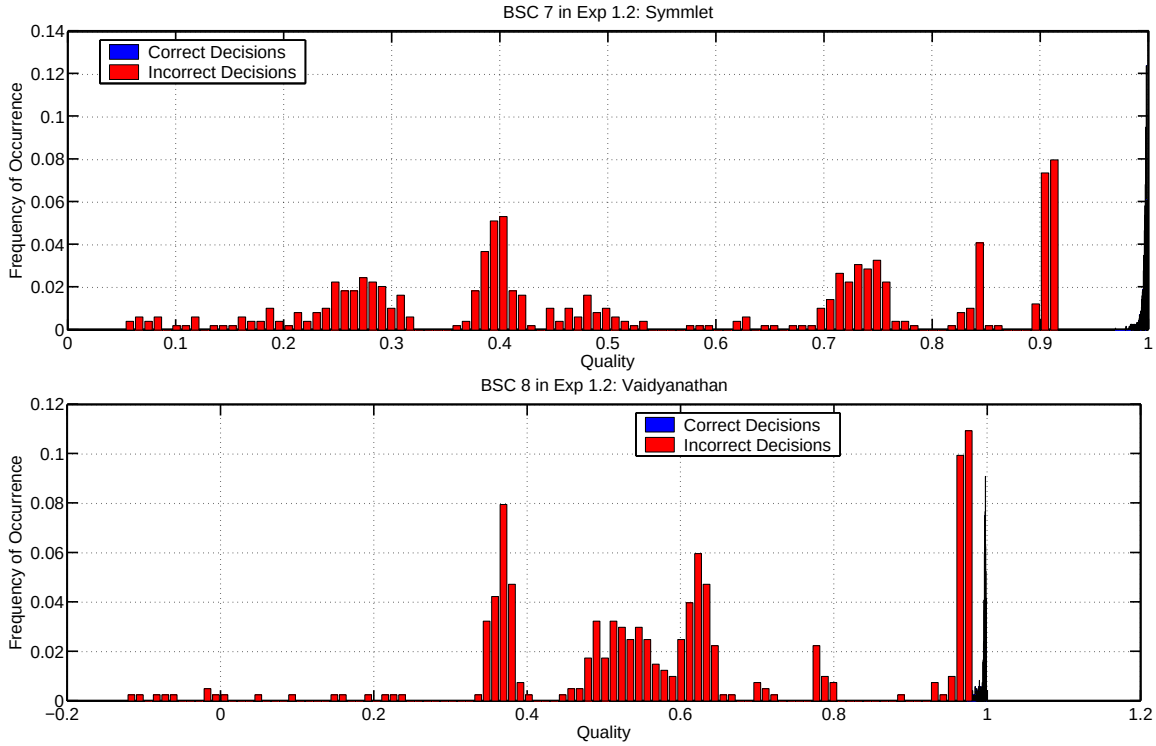


Figure 65: Quality histograms for BSCs 7 and 8 in Experiment 1.2.

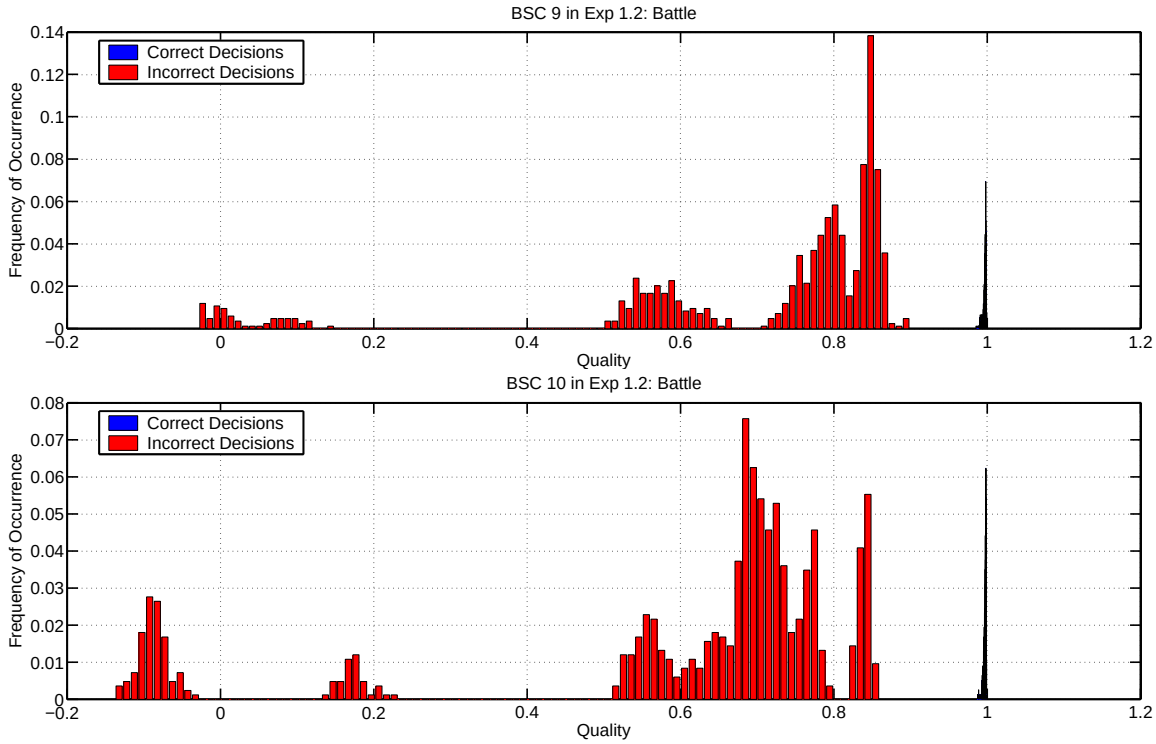


Figure 66: Quality histograms for BSCs 9 and 10 in Experiment 1.2.

5.1.3 Experiment 1.3: Path Correction in the BTC

In this experiment, we revisit Experiment 1.1 but we use path correction in the BTCs and the BSC. Path correction has not be implemented for the BHC. Recall that path correction allows a BTC to follow several different paths through the tree until one is found that has sufficiently high decision quality. The paths are selected by detecting the presence of a decision node with poor quality and retraversing the tree and forcing this node to make the alternate choice. That is, we do not simply compute all paths through the tree in an exhaustive way and pick the best. In this way, path correction usually involves a small number of iterations.

Probability of Correct Classification The probability of correct classification for Experiment 1.3 is shown in Figure 67 (compare to Figure 10). Note that the performances for all BTCs and the BSC are improved over the corresponding performance for Experiment 1.1. The most dramatic increase is for CID 3, which has a probability of correct classification of 0.4 without path correction and greater than 0.9 with path correction.

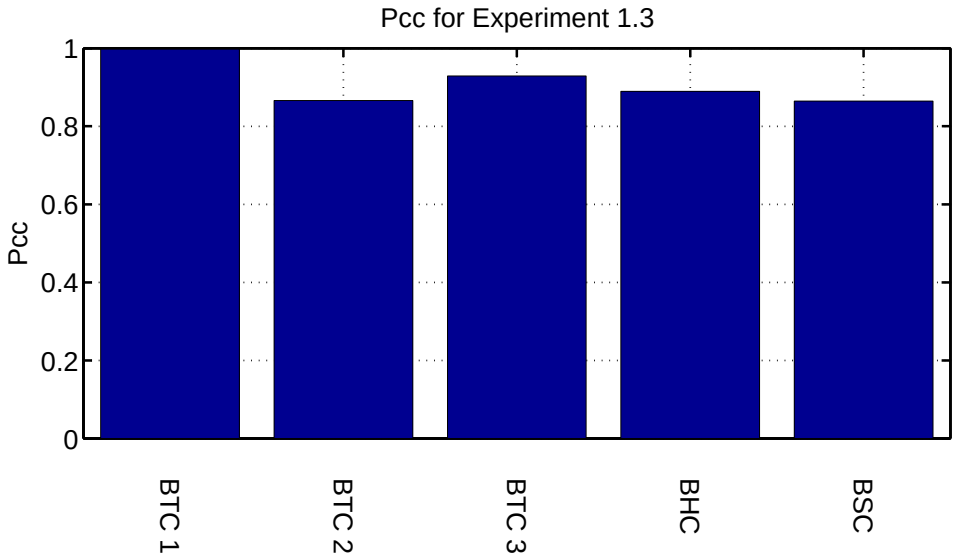


Figure 67: Classification performance for Experiment 1.3

Confusion Matrices The confusion matrices for Experiment 1.3 are shown in Figures 68–70. Note that for BTCs 1 and 3 and for the BSC there are only a few ambiguous classes. Performance for BTC 1, which directly operates on the MLSR sequence, is near perfect, indicating that the automatically obtained tree structure together with path correction can find the small number of wavelet basis vectors needed for excellent performance with no prior knowledge provided to the algorithm.

Quality Measures The decision-quality histograms for Experiment 1.3 are shown in Figures 71–75. There are much fewer low-quality incorrect decisions as compared to Experiment 1.1 (no path correction). However, the BHC, which does not yet have path correction available, still produces a large number of low-quality incorrect decisions.

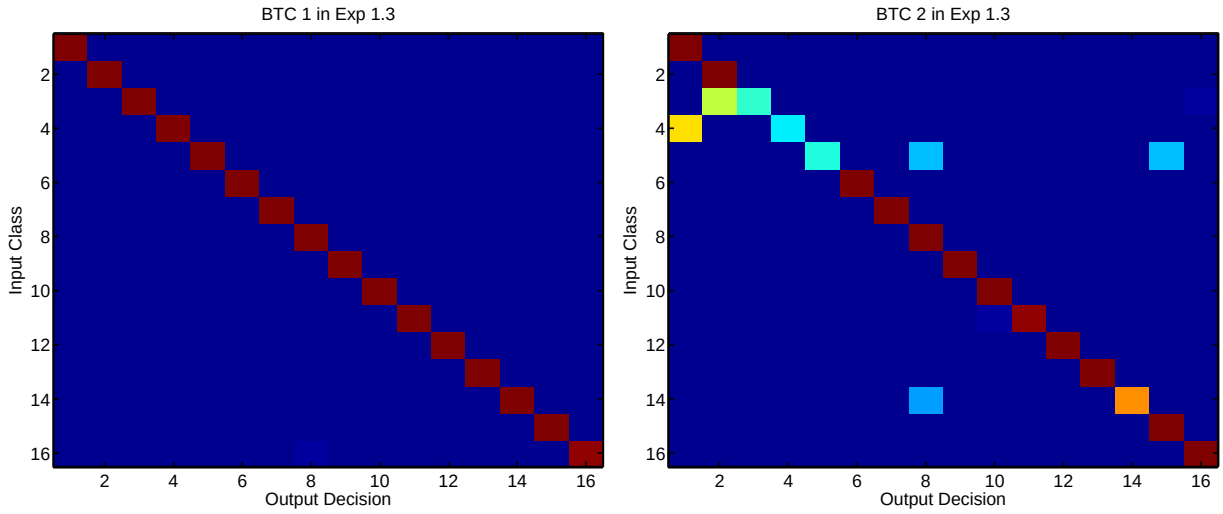


Figure 68: Confusion matrices for BTCs 1 and 2 in Experiment 1.3.

Conclusions for Experiment 1.3

1. Path correction appears to substantially improve performance for a BTC (or BSC, being a special case of the BTC). For CID 3, performance improved from $P_{cc} = 0.4$ to 0.9.
2. The presence of a large number of very low-quality decisions for the BHC indicates that the BHC may also be greatly improved by employing some form of path correction. This has not yet been done since the path through the hypertree is much more complex than the path through a single BTC.
3. Our suspicion that path correction is particularly well suited to problems having soft class ambiguities is supported by the results of Experiment 1.3. In Experiment 2 we will look for further evidence, since the classes in that problem have multiple hard ambiguities.

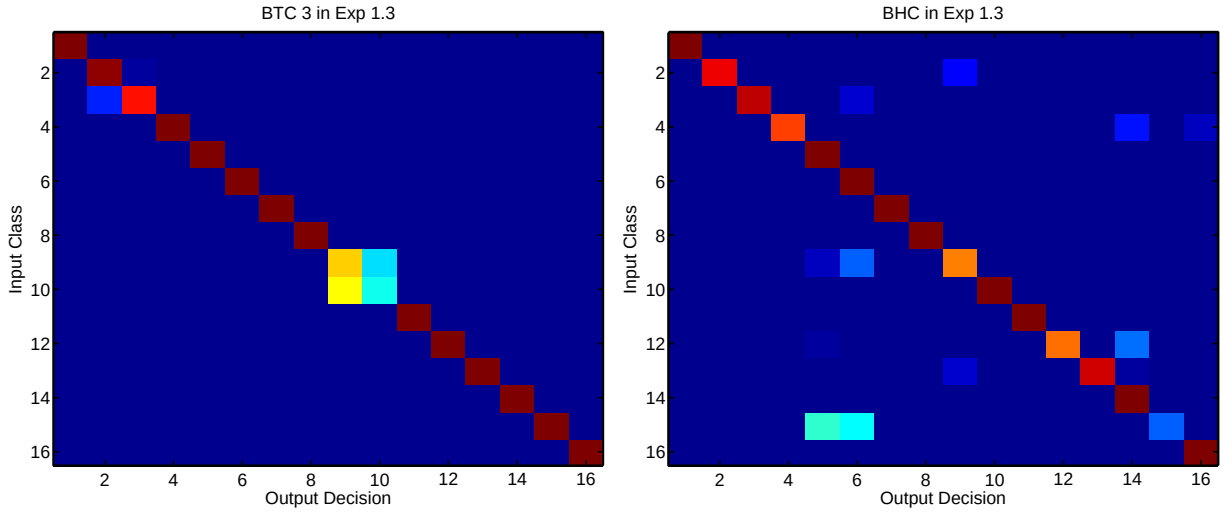


Figure 69: Confusion matrices for BTC 3 and the BHC in Experiment 1.3.

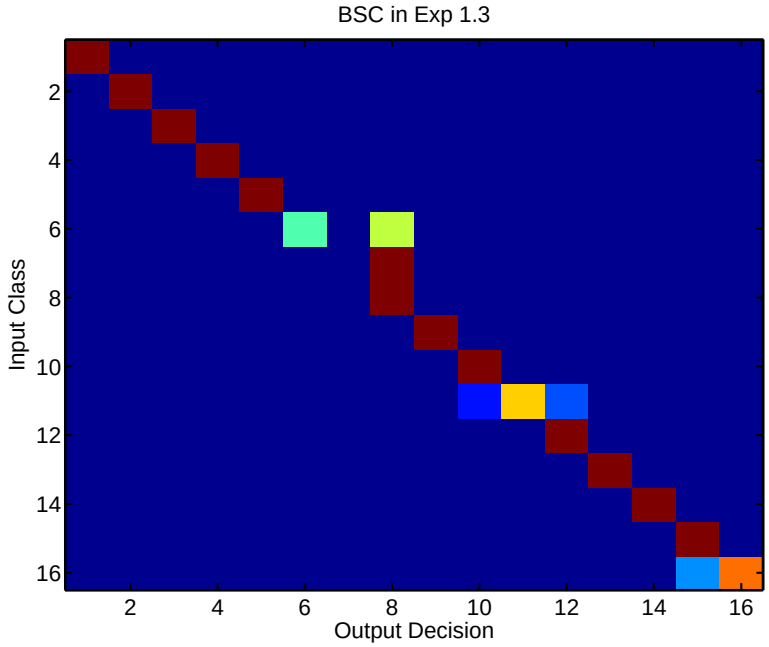


Figure 70: Confusion matrix for the BSC in Experiment 1.3.

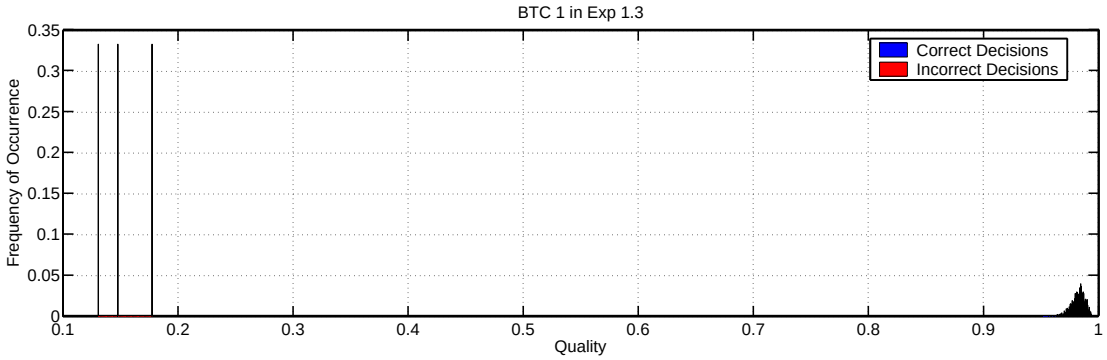


Figure 71: Quality histogram for BTC 1 in Experiment 1.3.

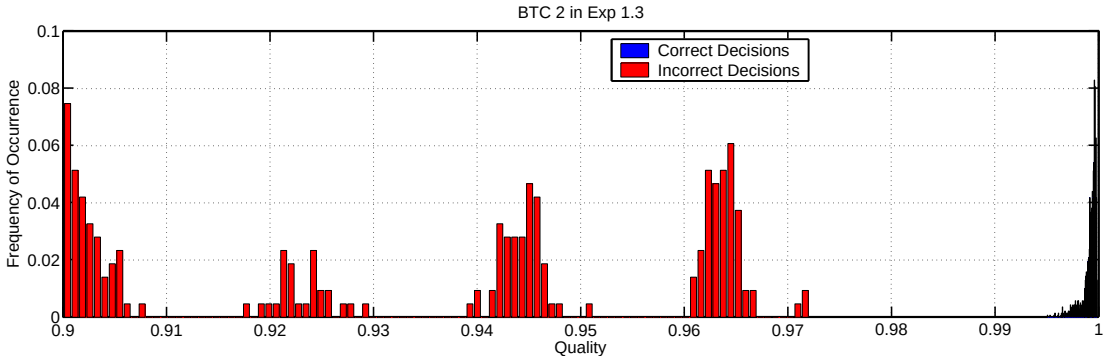


Figure 72: Quality histogram for BTC 2 in Experiment 1.3.

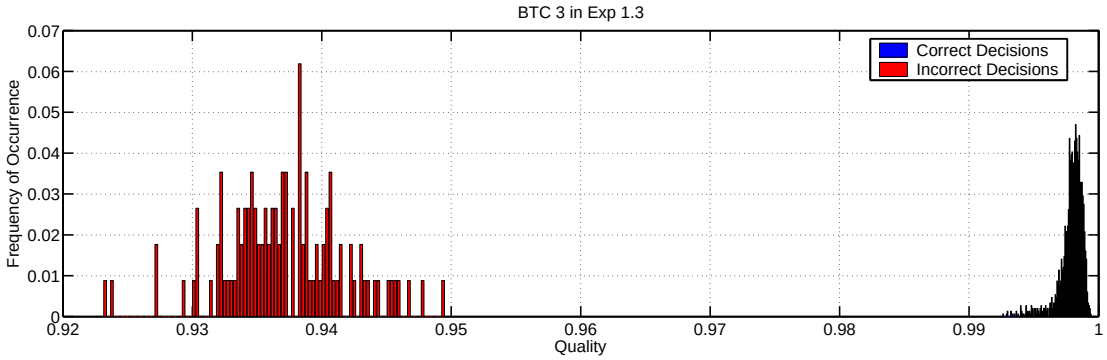


Figure 73: Quality histogram for BTC 3 in Experiment 1.3.

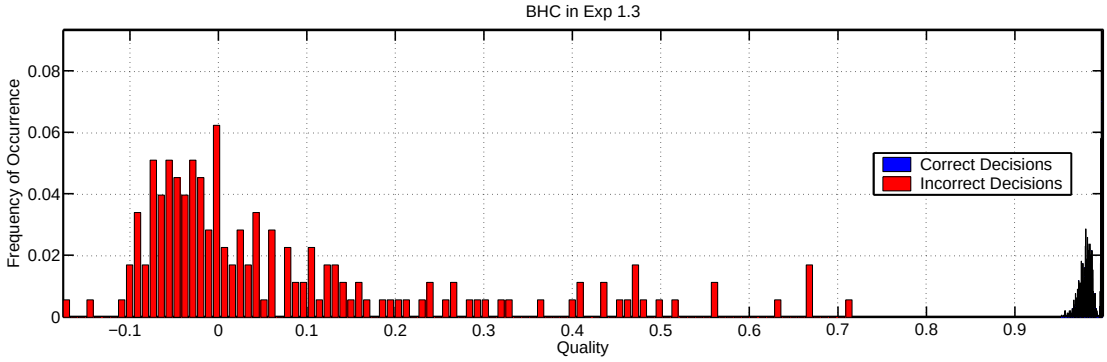


Figure 74: Quality histogram for the BHC in Experiment 1.3.

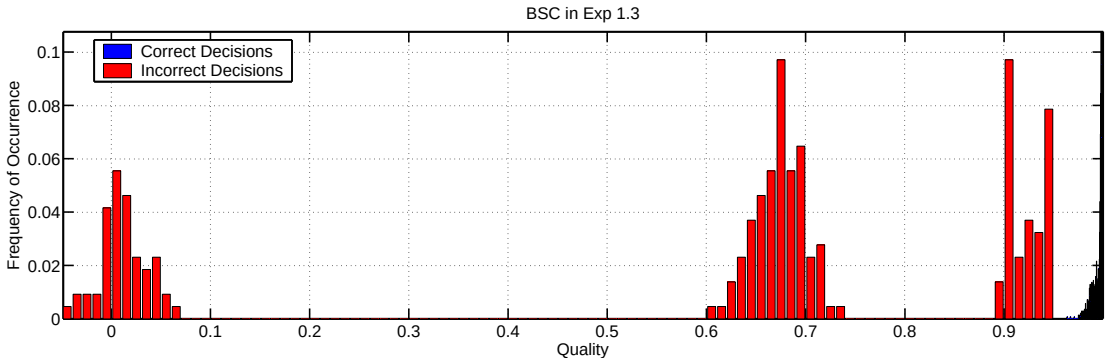


Figure 75: Quality histogram for the BSC in Experiment 1.3.

5.2 Toy Problem Two: Two-Dimensional Inputs

In this section, we report on a number of related classification experiments for which the synthetic input classes consist of eight objects. The objects are represented as images (two-dimensional inputs), and there are four classifier input data (CID) types. The physical interpretation is that each object can be viewed through one of four distinct camera types. The images have dimension 32 by 32 for each CID. The goal of the processing is to correctly classify the object type (decide on the class label) given a minimum number of CIDs.

Classifier Input Data Types.
The eight classes and four CIDs result in the thirty-two images shown in Figure 76. The eight underlying abstract objects are seen through four cameras: black-and-white, gray-scale, color, and infrared. Notice the substantial ambiguity built into each of the CIDs. For example, for CID 1 (black-and-white camera), classes 1, 2, 5, and 6 are identical, and classes 7 and 8 are identical. So the best possible classifier for this type of CID will produce many errors. The idea behind the present research is the development of a classifier that can take advantage of the additional CIDs in such a way as to resolve all ambiguities while simultaneously using the minimum number of additional CIDs.

Experimental Outputs.
The outputs of the experiments are identical to those in Experiment 1: the obtained classification trees, the probability of correct classification for each classifier, the confusion matrix for each classifier, quality-measure histograms, and the number of distinct CIDs required by the BHC.

5.2.1 Experiment 2.1: Basic 2-D Processing

In the first two-dimensional experiment, the BTC, BHC, and BSC structures are obtained for a single wavelet type and high SNR. Classification experiments are performed to illustrate the relative effectiveness of the BTCs, the BHC, and the BSC. This establishes the basic correctness of the approach and code, and sets up a baseline for comparison with more complex experimental results. The parameters for Experiment 2.1 are listed in Table 3.

Automatically Obtained BTCs and BSC The trees for the obtained BTCs and the BSC are shown in Figures 77–85. Note that the first four BTCs are the basic BTCs, and the remainder are derived (see Section 5.1). It is instructive to study the four basic BTCs while keeping in mind the ambiguities shown in Figure 76. This is done in the following paragraphs.

For CID 1 (black-and-white camera), we have the BTC in Figure 77. The important point is that the ambiguities obvious from Figure 76 are reflected in the tree structure. In particular, the ambiguous classes of 7 and 8 are assigned their own branch in the tree, as are the four-way ambiguous classes 1,2,5, and 6. Therefore, classes 7 and 8 will be confused for each other, as is natural, and 1,2,5, and 6 will be confused for each other. If the classes 7 and 8 are replaced by class A and classes 1,2,5, and 6 are replaced by the single class B, then the resulting tree would have the structure required by the formalism of this work: there would be a single path through the tree to A and a single path to B. In this sense, the obtained tree should produce perfect results.

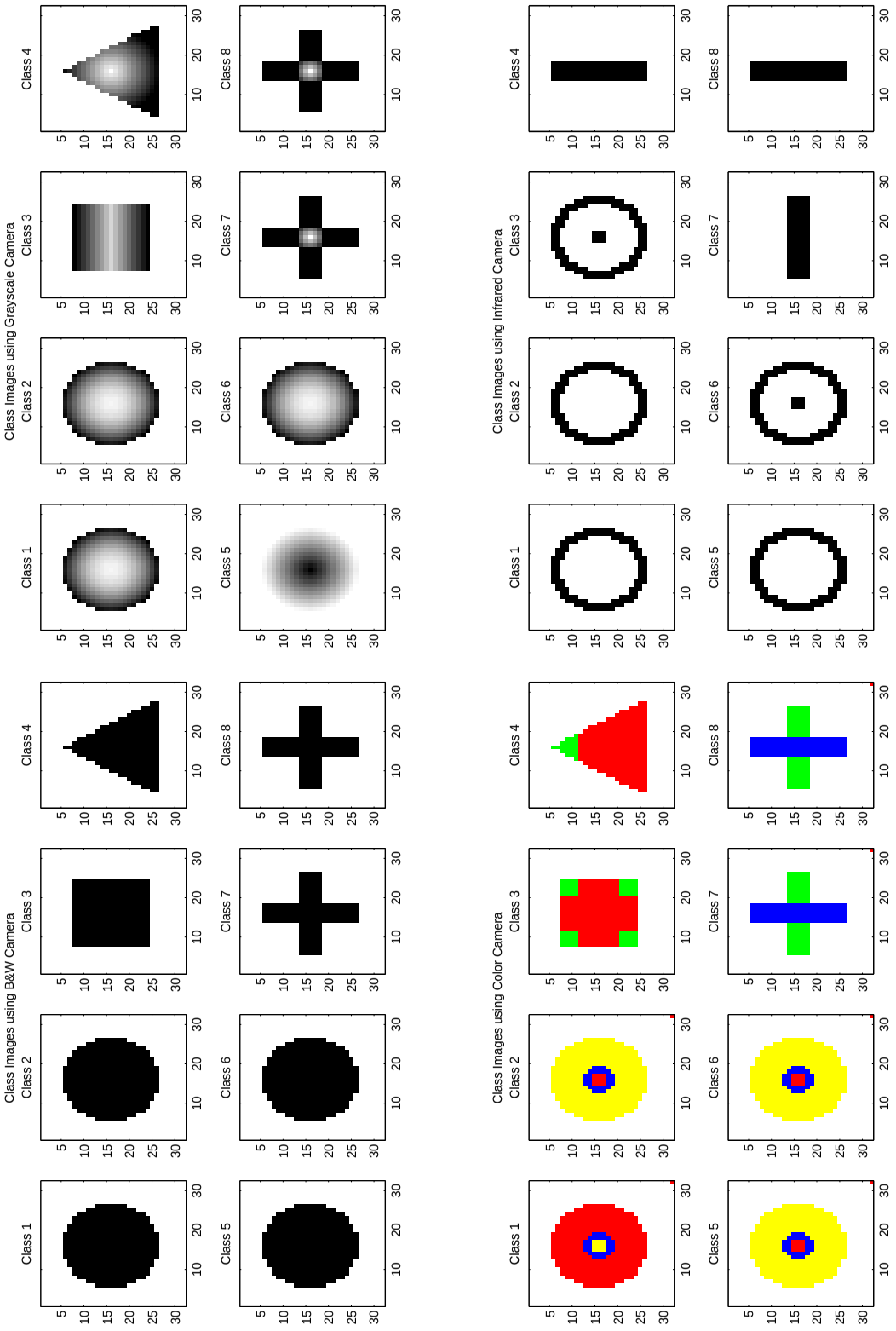


Figure 76: The eight classes for the 2D experiments, seen through each of the four CID types.

Parameter	Value
Wavelet Type	Coiflet
Wavelet Parameter	5
Feature Length K	20
Number of Classes C	8
BTC/BHC Wavelet Tree Depth J	4
BSC Wavelet Tree Depth J	5
Number of CIDs	4
Data Dimension	[32 32]
Training SNR	∞
Input SNR CIDs 1,2,3,4	10dB
Random Translation	None
Random Scaling	None
Tree Topology	Free
Superclass Assignment	Free
Number of Trials	100

Table 3: Experimental parameters for the first 2-D experiment.

For CID 2 (gray-scale camera), we have the BTC in Figure 78. Again, the inherent ambiguities are reflected in the structure. Here the classes 7 and 8 form one ambiguous set, and the classes 1,2, and 6 form the other.

For CID 3 (color camera), the BTC is shown in Figure 79. Here the obtained structure does not quite reflect the inherent ambiguity of the problem. Classes 7 and 8, which form an ambiguous class, are found in separate regions of the tree. The other set of ambiguous classes, 2, 5 and 6, are successfully isolated and grouped together in the tree.

Finally, for CID 4 (infrared camera), the BTC is given in Figure 80. All ambiguous sets are correctly represented in the tree: $\{1, 2, 5\}$, $\{4, 8\}$, and $\{3, 6\}$.

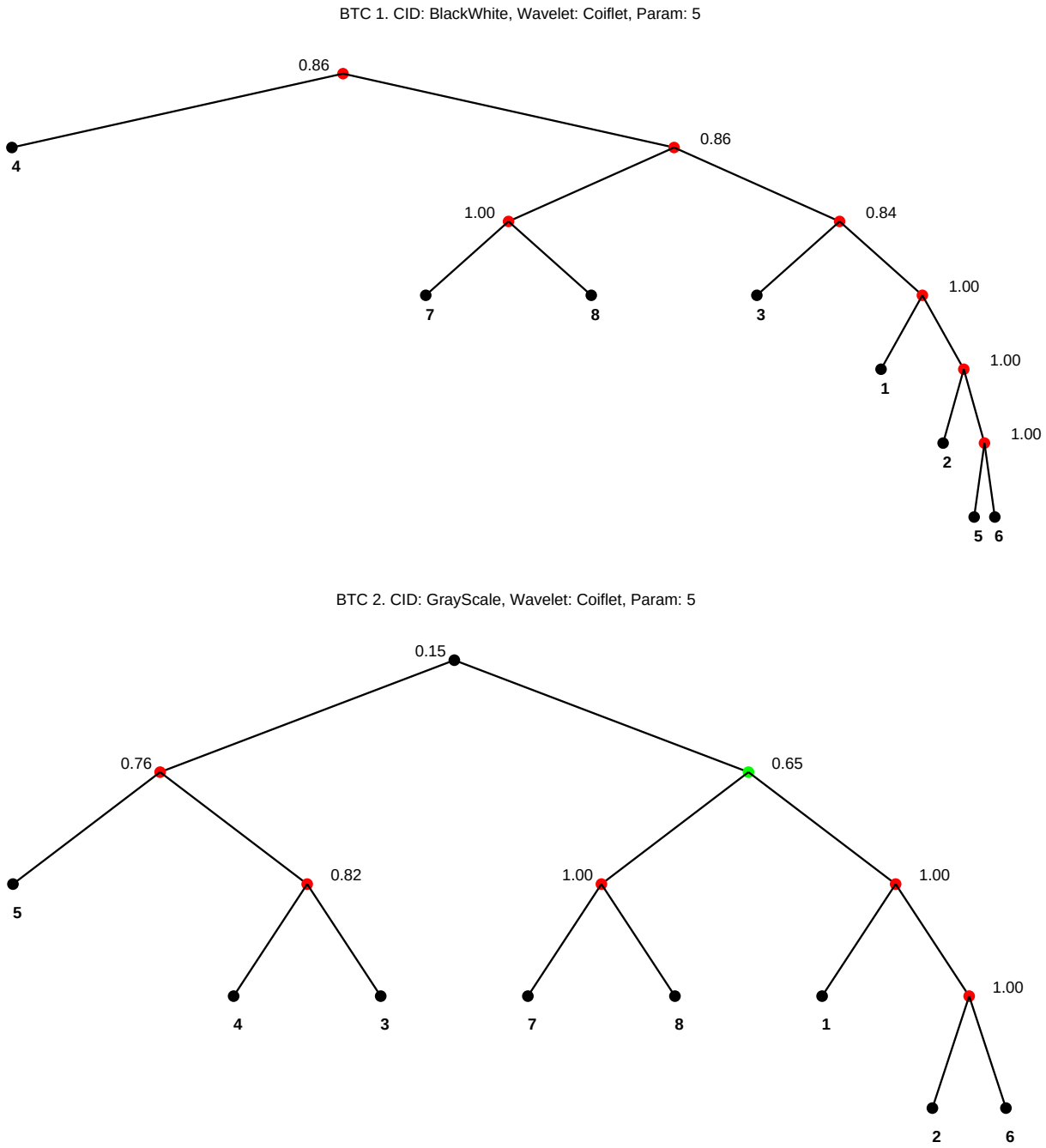


Figure 77: BTCs obtained for Experiment 2 (1–2 of 16).

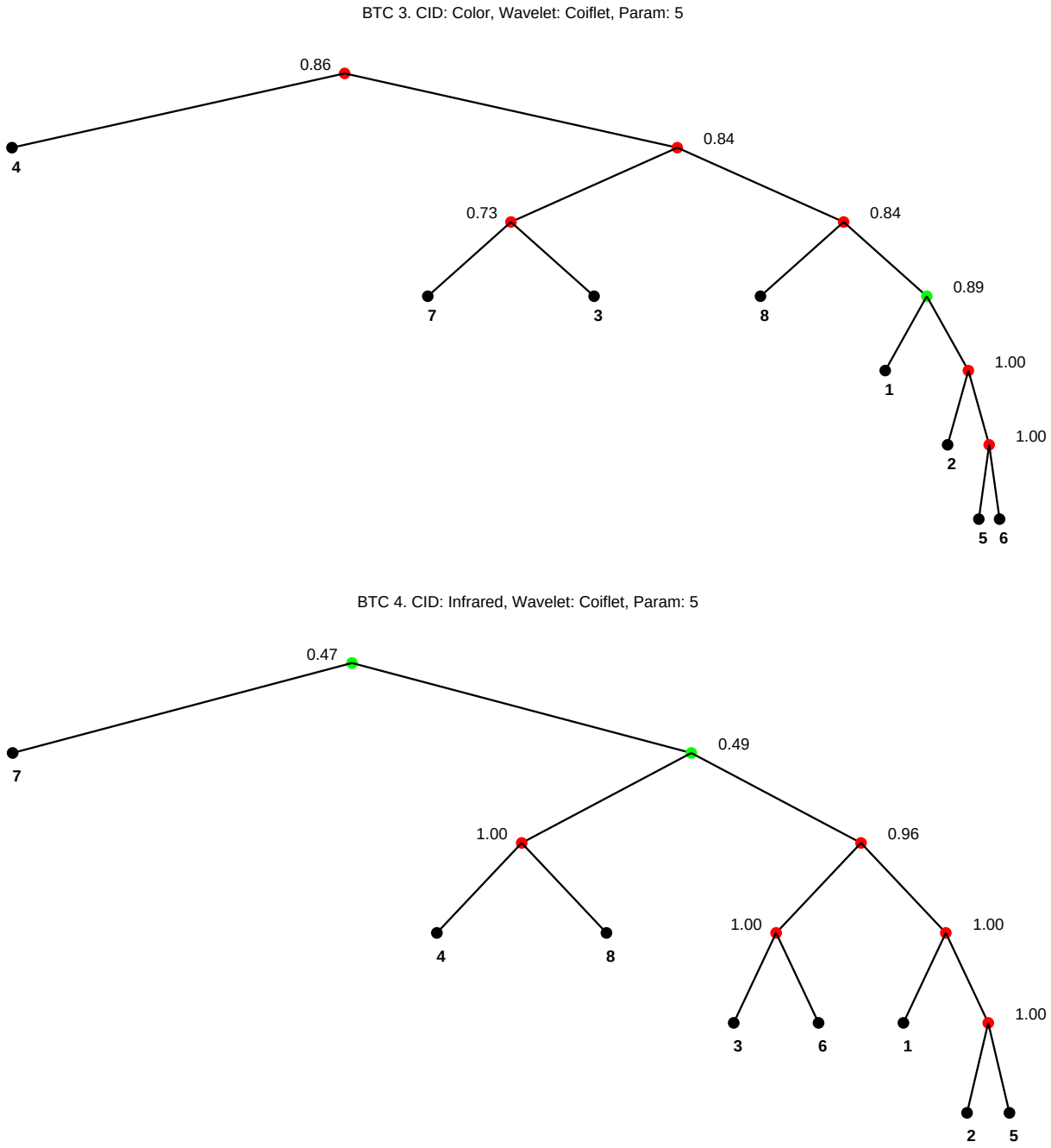


Figure 78: BTCs obtained for Experiment 2 (3–4 of 16).

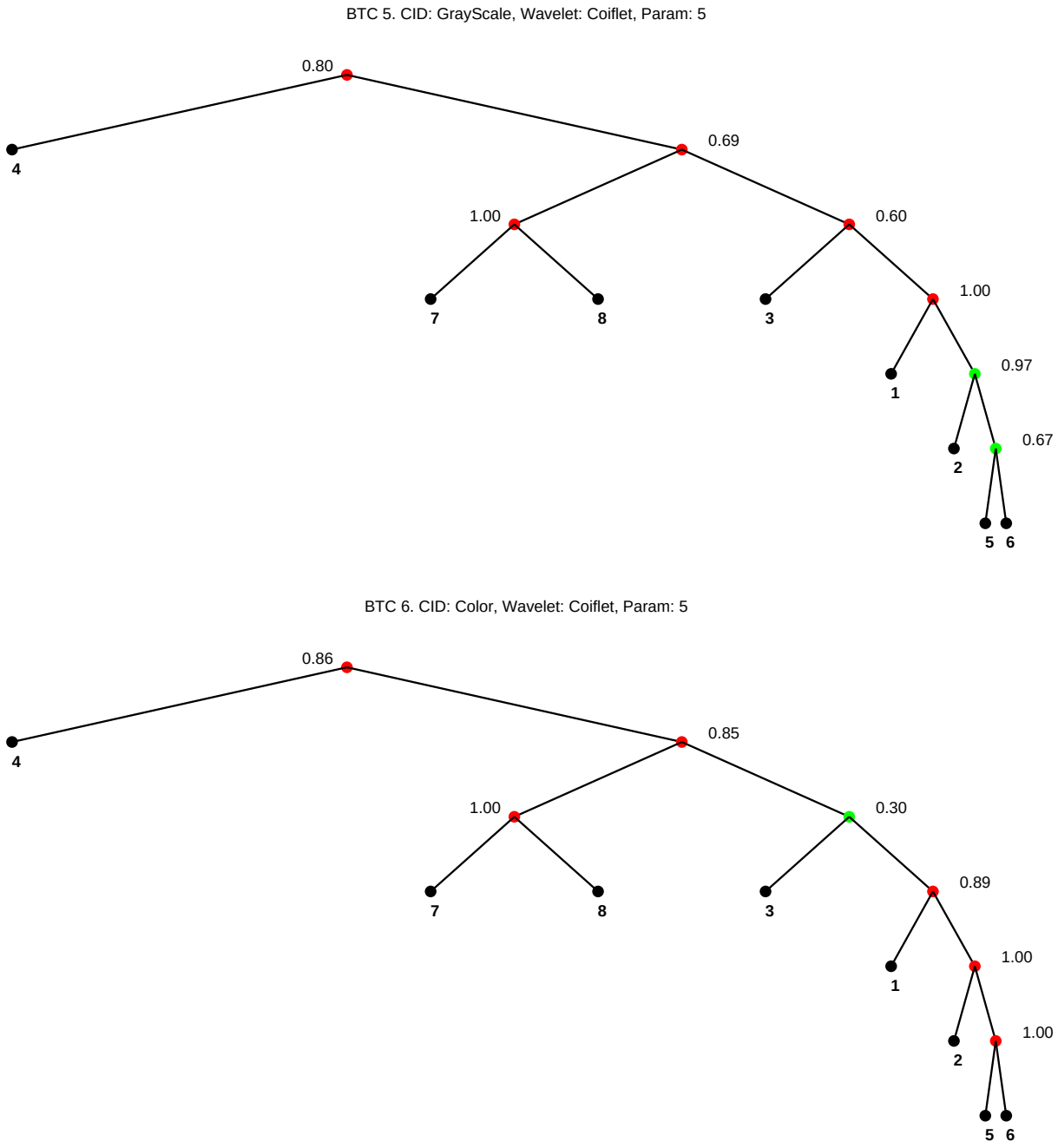


Figure 79: BTCs obtained for Experiment 2 (5–6 of 16).

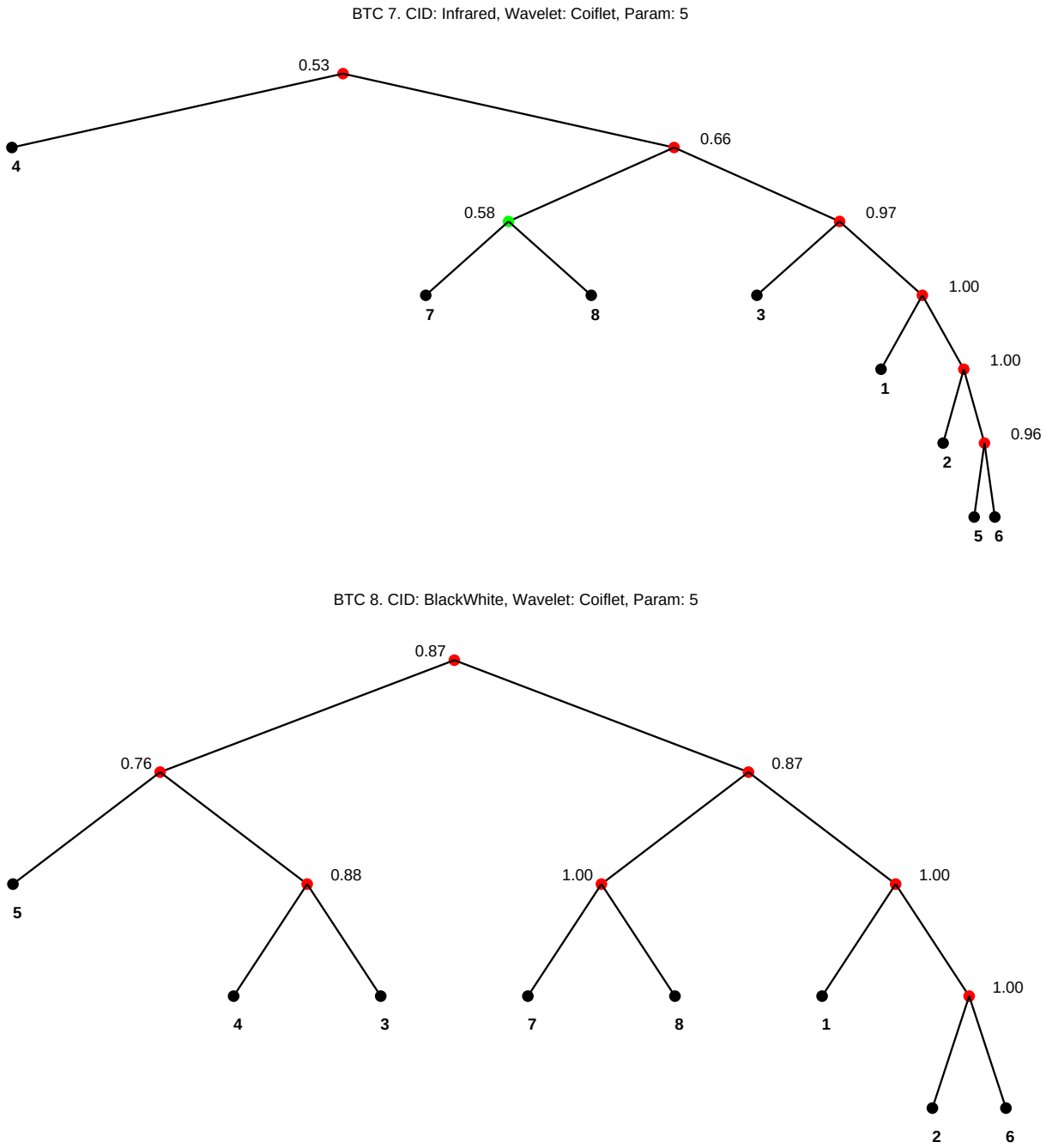


Figure 80: BTCs obtained for Experiment 2 (7–8 of 16).

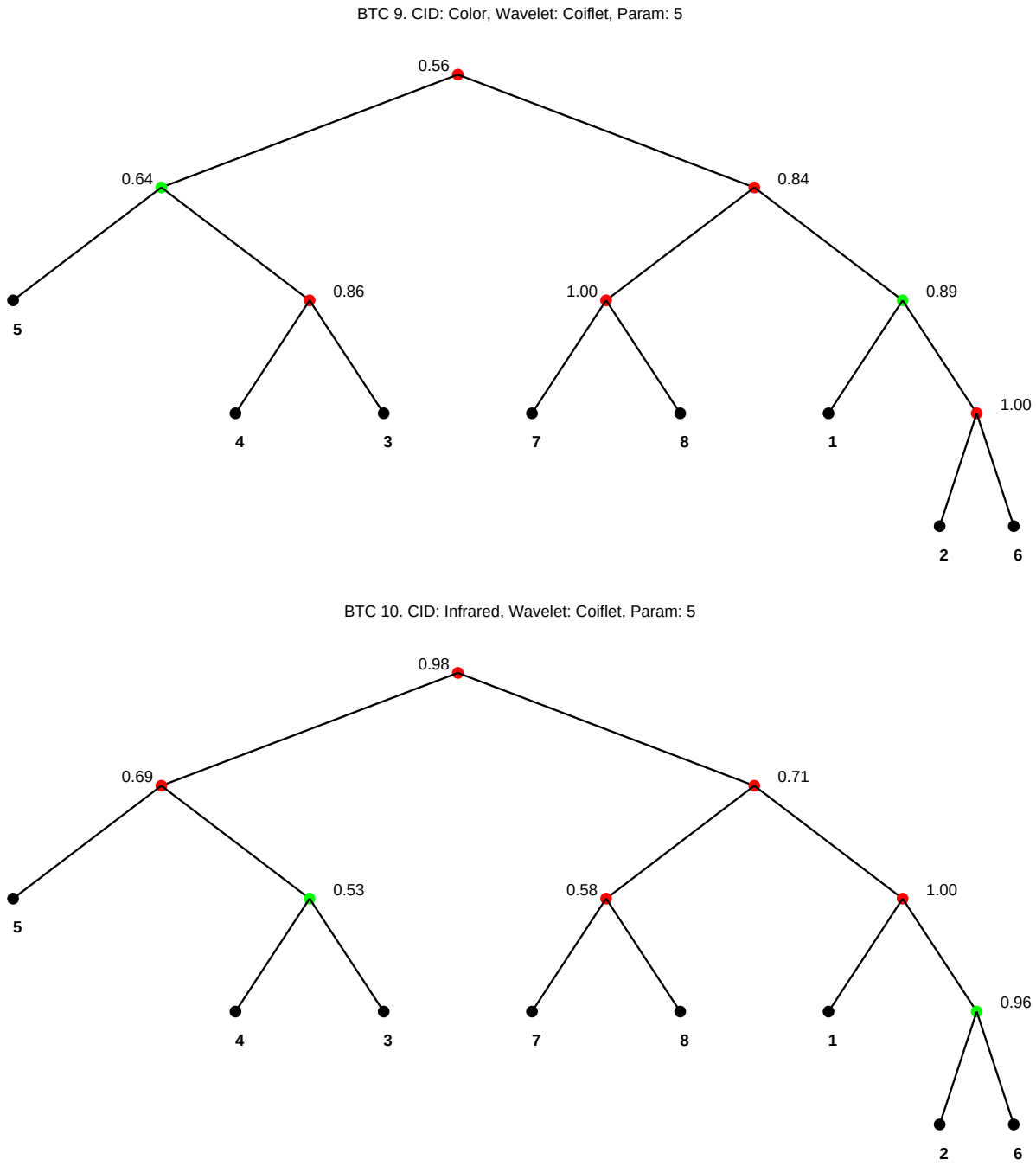


Figure 81: BTCs obtained for Experiment 2 (9–10 of 16).

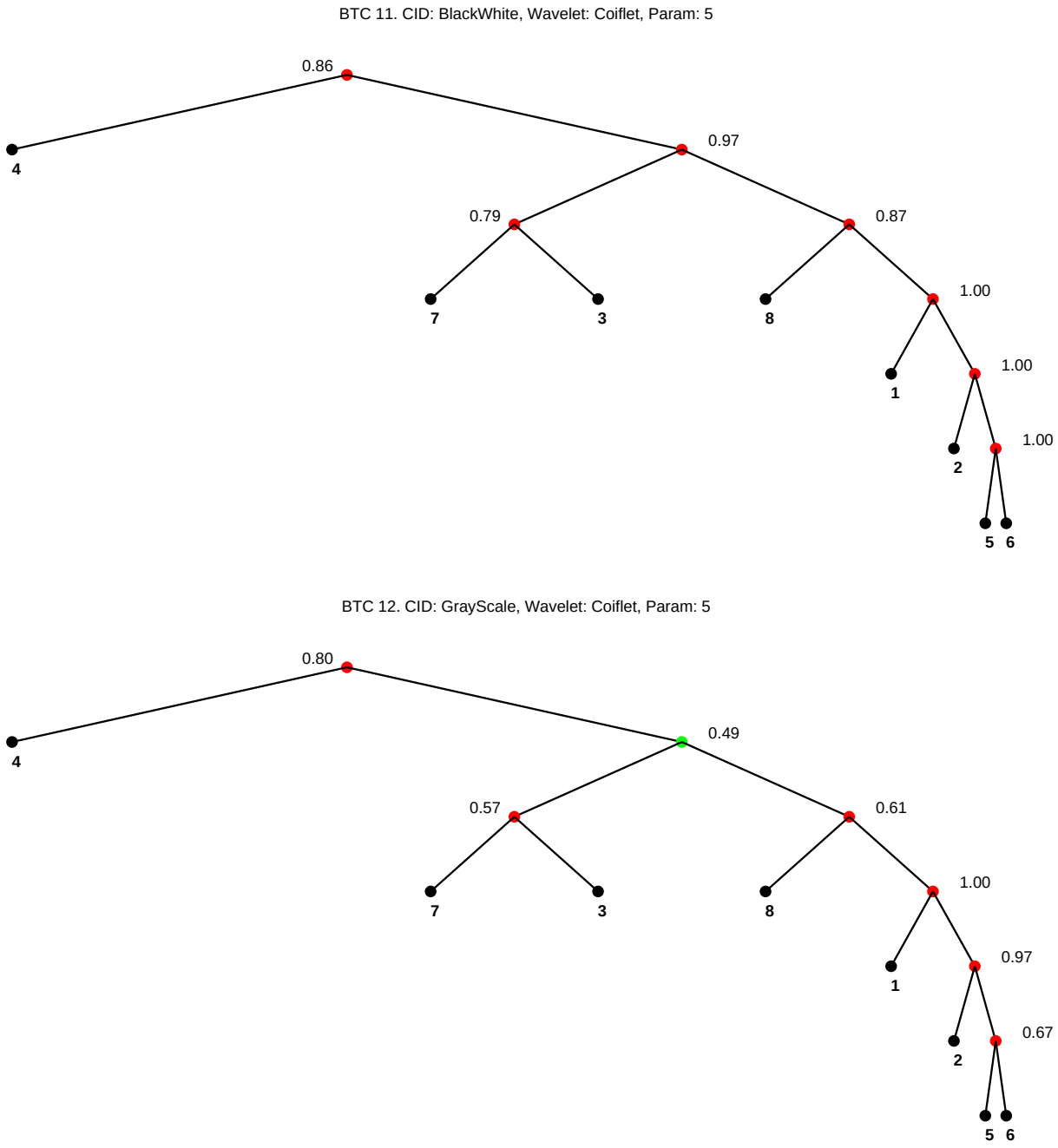


Figure 82: BTCs obtained for Experiment 2 (11–12 of 16).

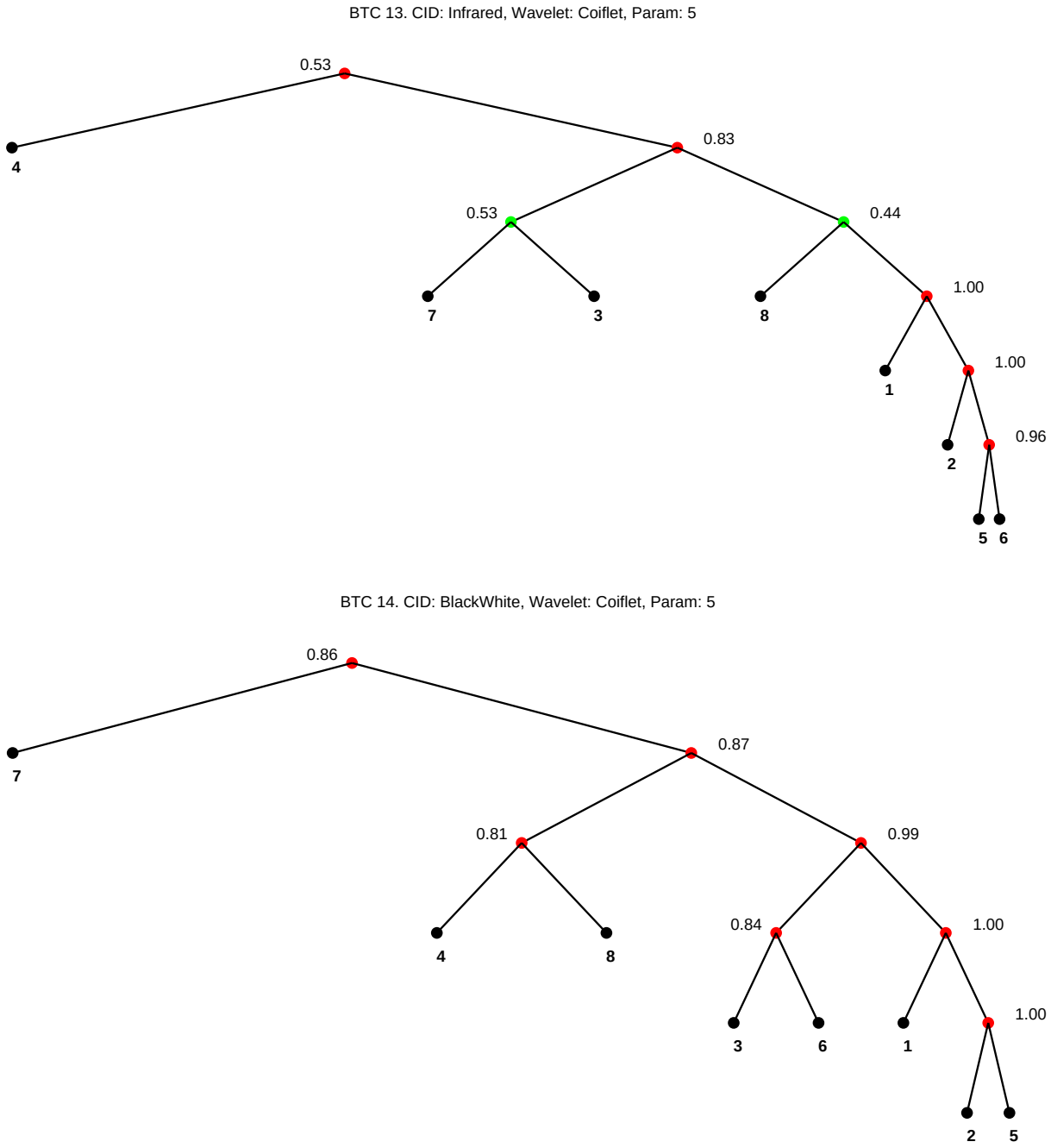
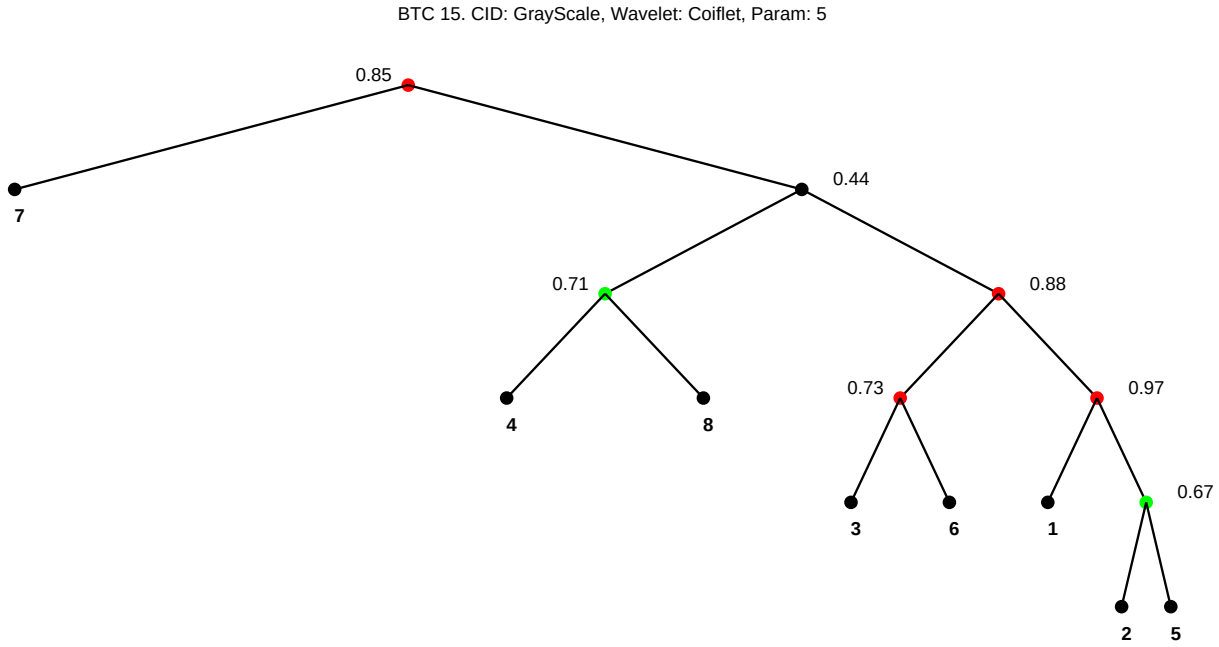


Figure 83: BTCs obtained for Experiment 2 (13–14 of 16).



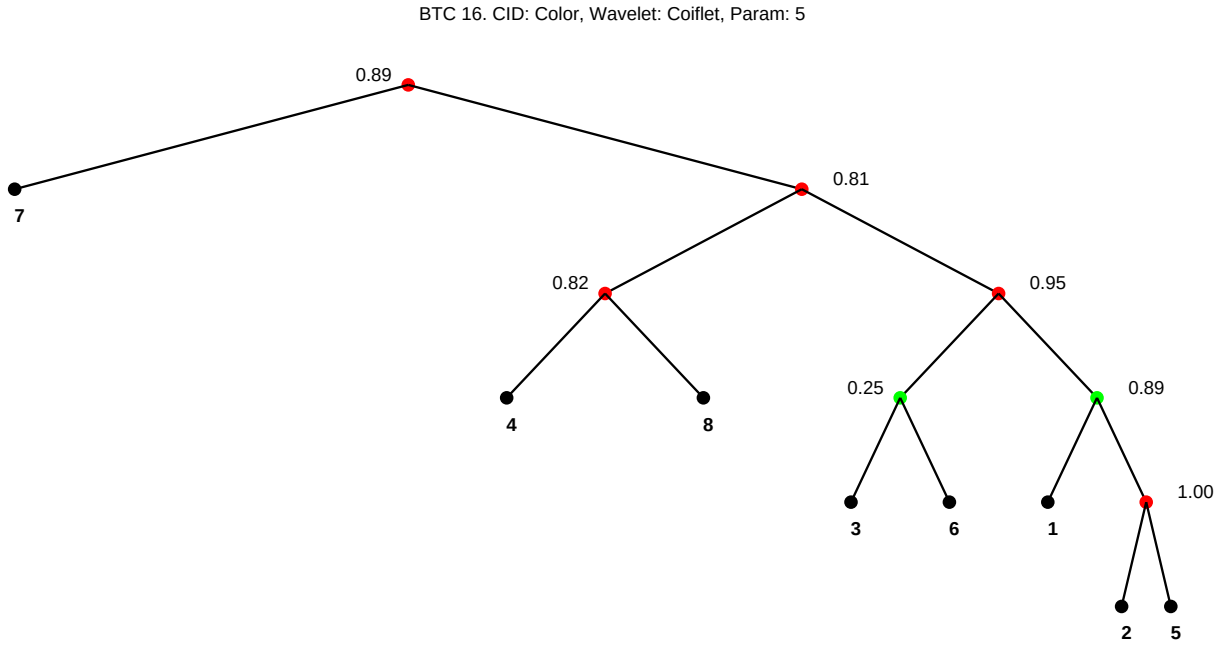


Figure 84: BTCs obtained for Experiment 2 (15–16 of 16).

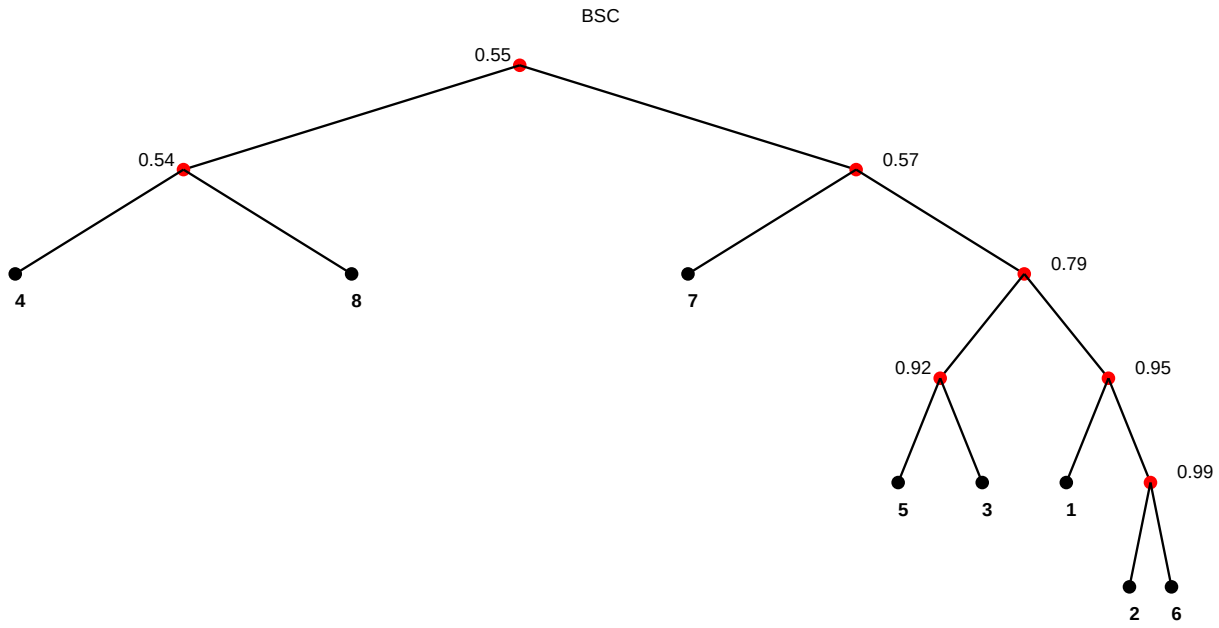


Figure 85: BSC obtained for Experiment 2.1.

Probability of Correct Classification The estimates of the probability of correct classification (P_{cc}) for Experiment 2.1 are given in Figure 86. Note that the predicted performance ordering holds: BSC $>$ BHC $>$ {BTC}.

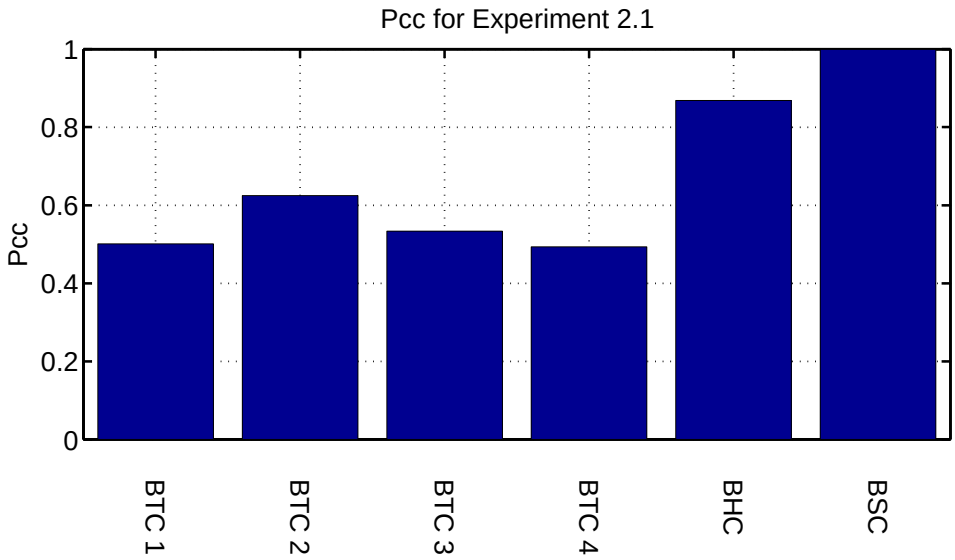


Figure 86: Probability of correct classification for Experiment 2.1.

Confusion Matrices The confusion matrices for the BTCs, BHC, and BSC for Experiment 2.1 are shown in Figures 87–89. It is very easy to see, for each CID, the correspondence between the ambiguous classes and the errors in the confusion matrix.

Quality Measures The histograms for the quality measure for Experiment 2.1 are shown in Figures 90–92. What is most notable here is the large number of very-high-quality incorrect decisions made in BTCs 1 and 3, and in the BHC. This should render path correction much less effective than in the one-dimensional experiments.

Conclusions for Experiment 2.1

1. The algorithms for creating basic binary tree classifiers and the hypertree classifier can work well even for problems involving multiple data types with severe ambiguities.
2. The performance ordering is as predicted by the theory, with the BSC outperforming the BHC, which outperforms all its constituent BTCs by a wide margin.
3. Excellent performance can be had through use of only a small number of key wavelet coefficients even when each consituent BTC in a hypertree has very poor performance.

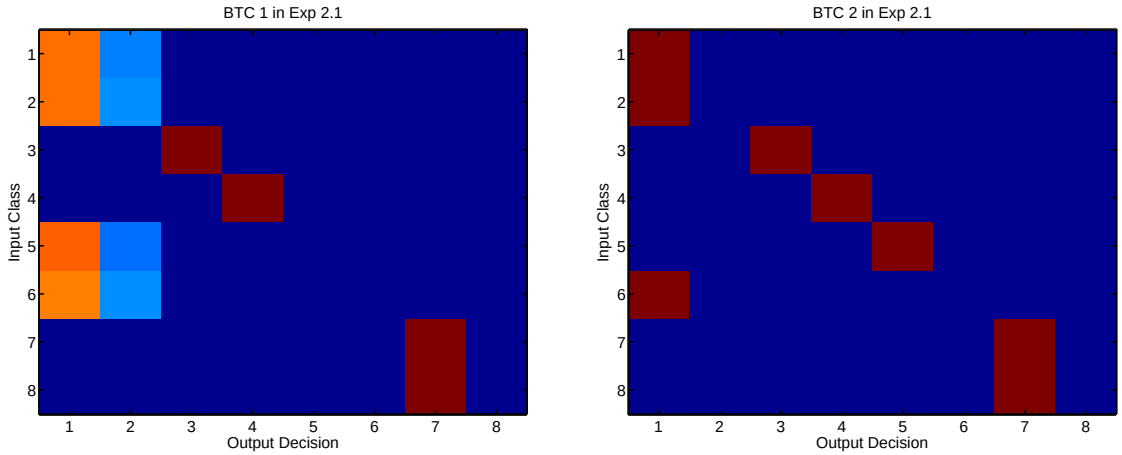


Figure 87: Confusion matrices for BTCs 1 and 2 in Experiment 2.1.

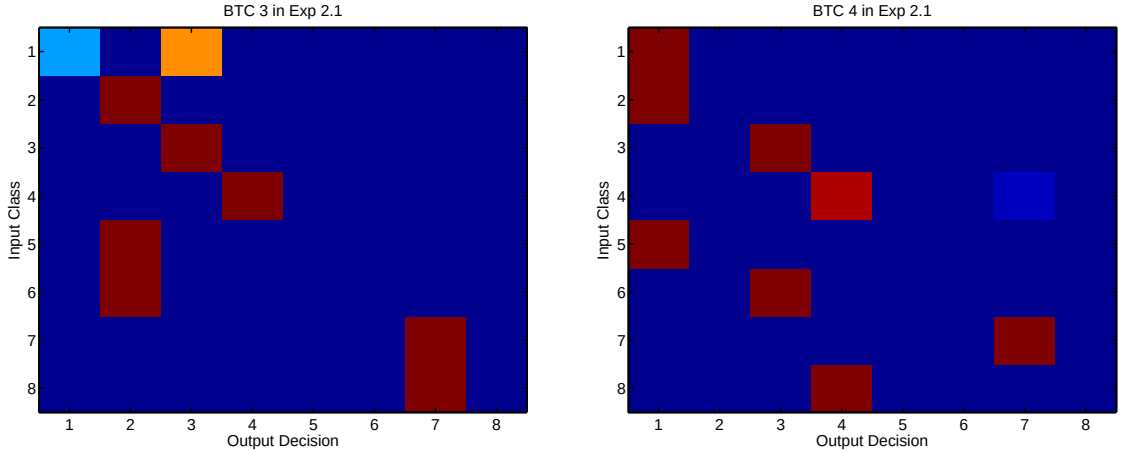


Figure 88: Confusion matrices for BTCs 3 and 4 in Experiment 2.1.

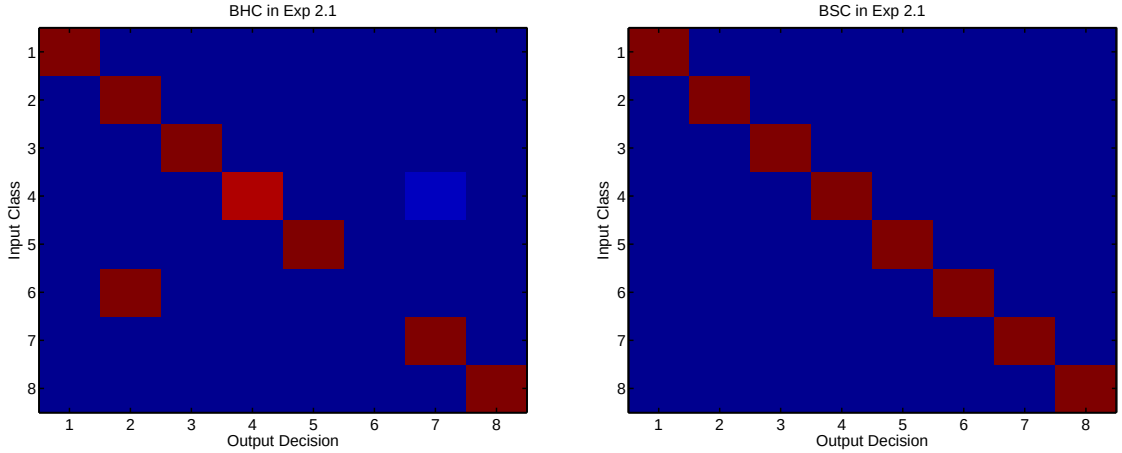


Figure 89: Confusion matrices for the BHC and BSC in Experiment 2.1.

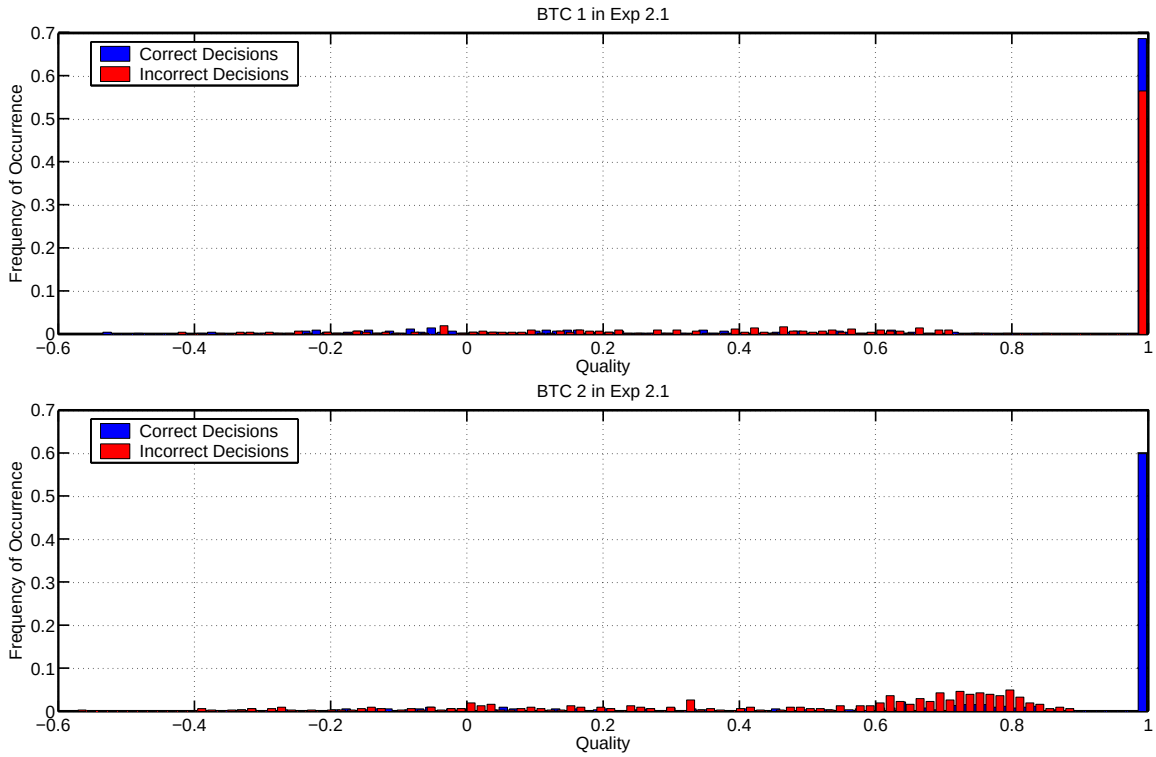


Figure 90: Quality histograms for BTCs 1 and 2 in Experiment 2.1.

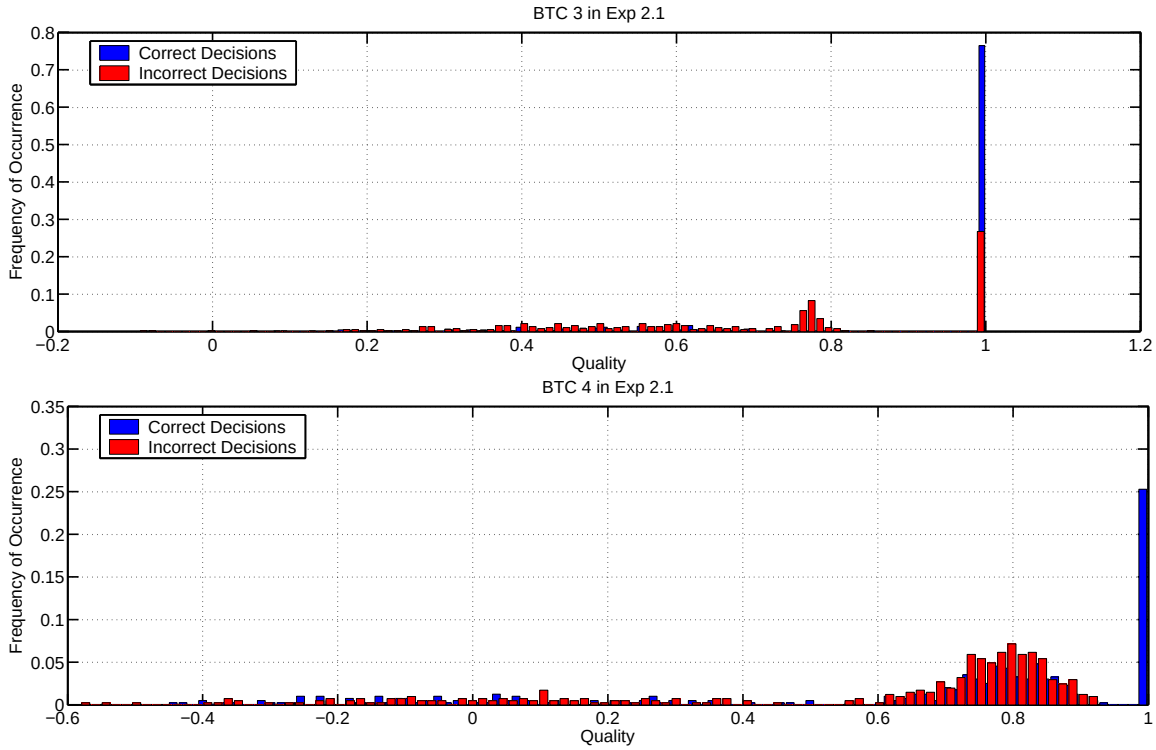


Figure 91: Quality histograms for BTCs 3 and 4 in Experiment 2.1.

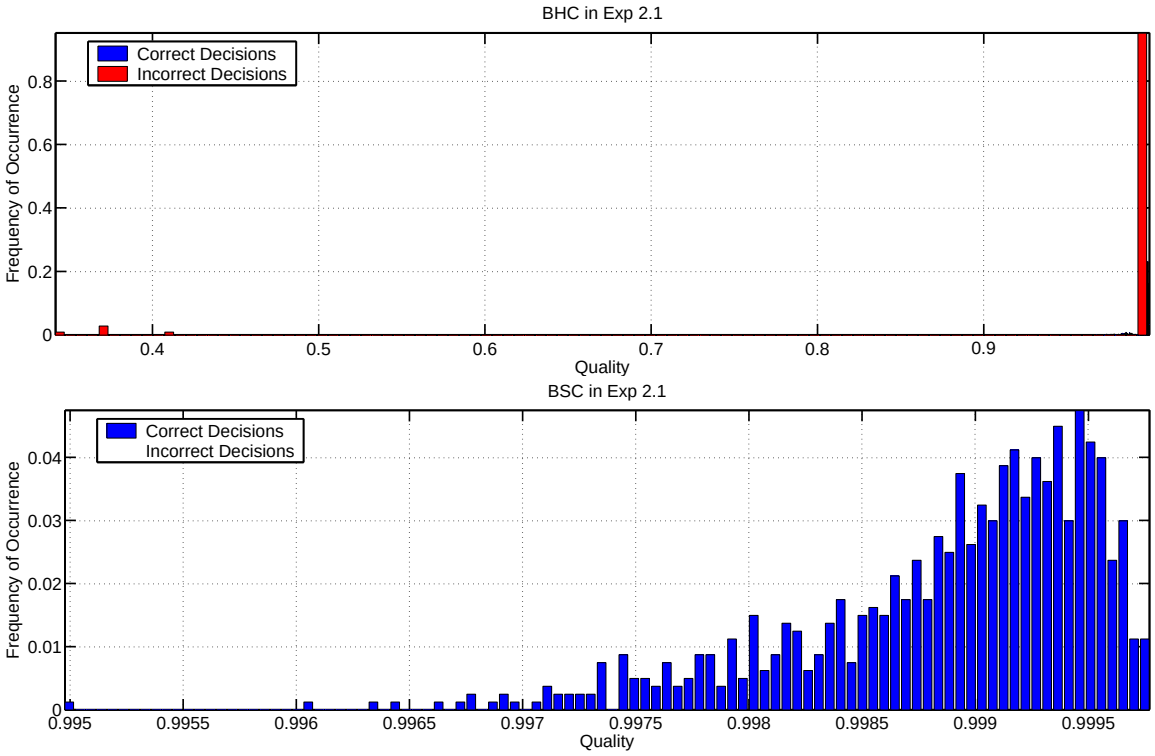


Figure 92: Quality histograms for the BHC and BSC in Experiment 2.1.

5.2.2 Experiment 2.2: Multiple Wavelets

In this experiment, we consider the influence of the particular wavelet used in forming the LDB (cf. Section 5.1.2). The parameters for the experiment are shown in Table 4. Six different wavelet types are employed and for three of these, two variants are used, for a total of ten distinct wavelets.

Parameter	Value
Wavelets	
Beylkin	
Coiflet	1
Coiflet	5
Daubechies	4
Daubechies	20
Symmlet	4
Symmlet	10
Vaidyanathan	
Battle	1
Battle	3
Feature Length K	20
Number of Classes C	8
BTC/BHC Wavelet Tree Depth J	4
BSC Wavelet Tree Depth J	5
Number of CIDs	4
Data Dimension	[32 32]
Training SNR	∞
Input SNR CIDs 1,2,3,4	10dB
Random Translation	None
Random Scaling	None
Tree Topology	Free
Superclass Assignment	Free
Number of Trials	100

Table 4: Experimental parameters for the first 2-D experiment.

Automatically Obtained BTCs and BSC The numerous automatically obtained BTCs are quite similar to those obtained in Experiment 2.1 and are not shown for reasons of brevity. The ten obtained BSCs are shown in Figures 93–97. All ten are distinct, but they reflect the sets of ambiguities inherent in the four CIDs, considered jointly (see Figure 76). For example, the classes 1, 2, 5, and 6 almost always appear together in an isolated part of the tree, which reflects the fact that either three or four of these classes are ambiguous in all CIDs. None of the BSCs have a particularly small ambiguity for the root node, reflecting again the fact that even the aggregate data (the four CIDs considered jointly) is riddled with ambiguity.

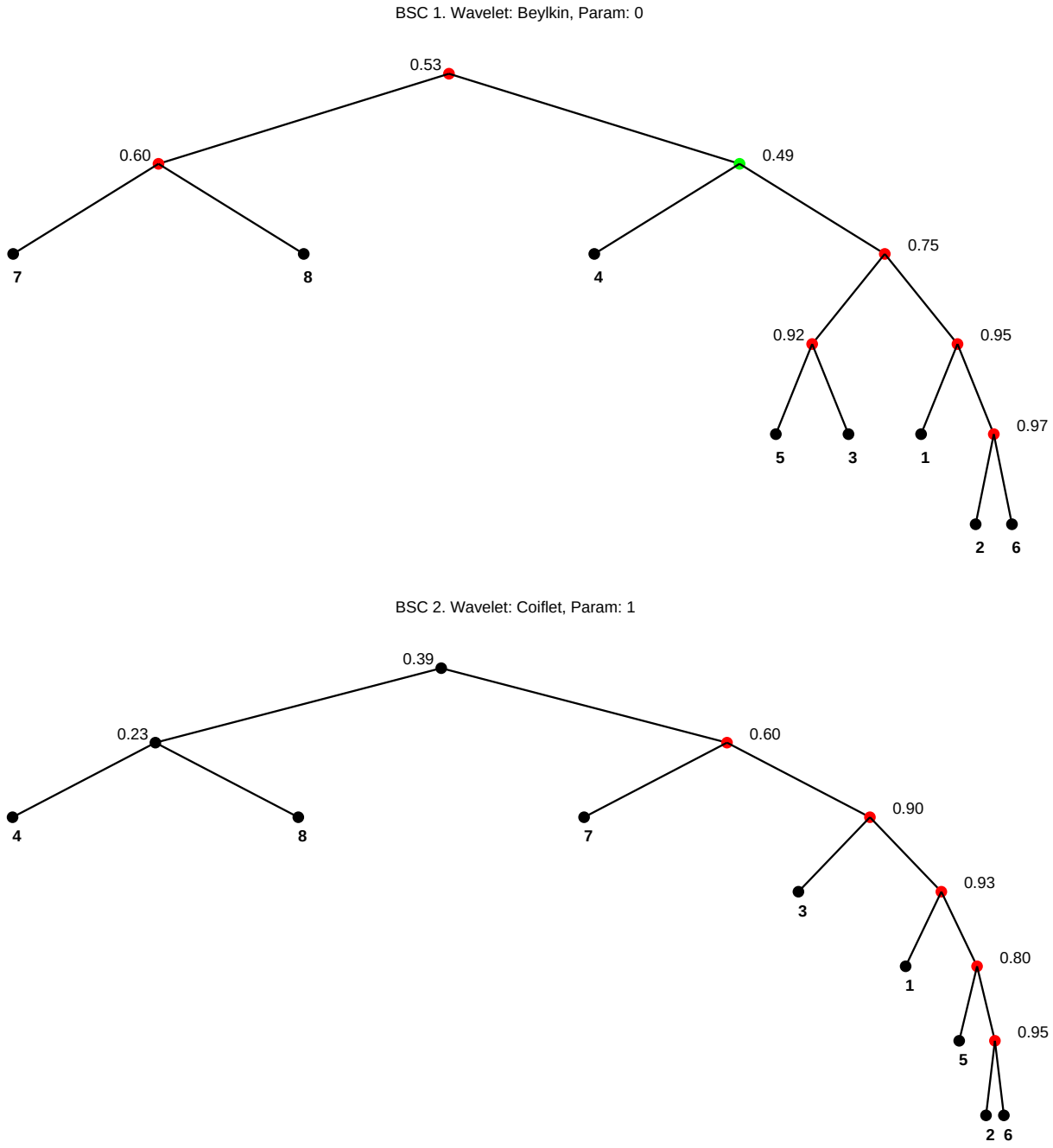


Figure 93: BSCs obtained for Experiment 2.2 (1–2 of 10).

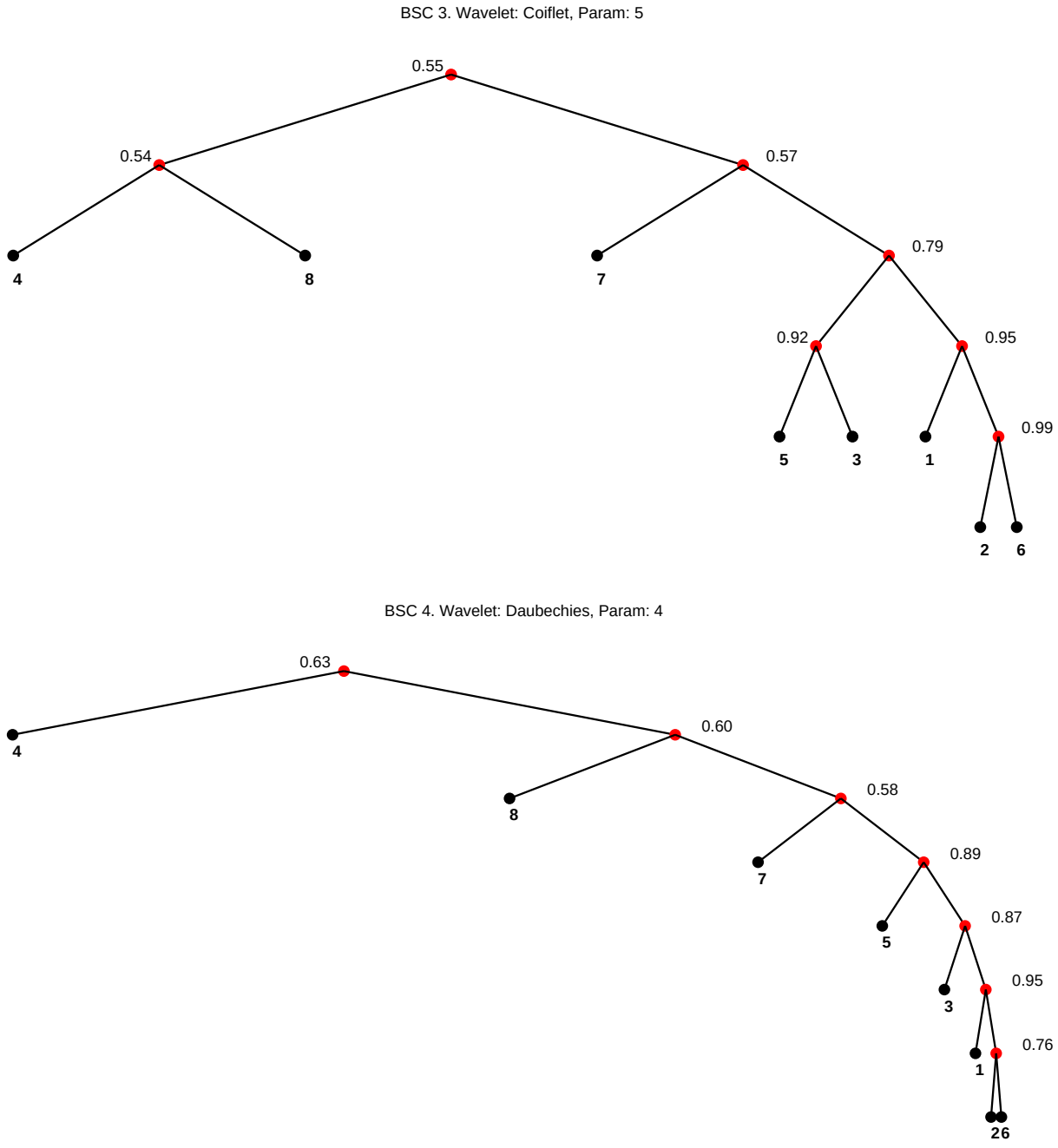


Figure 94: BSCs obtained for Experiment 2.2 (3–4 of 10).

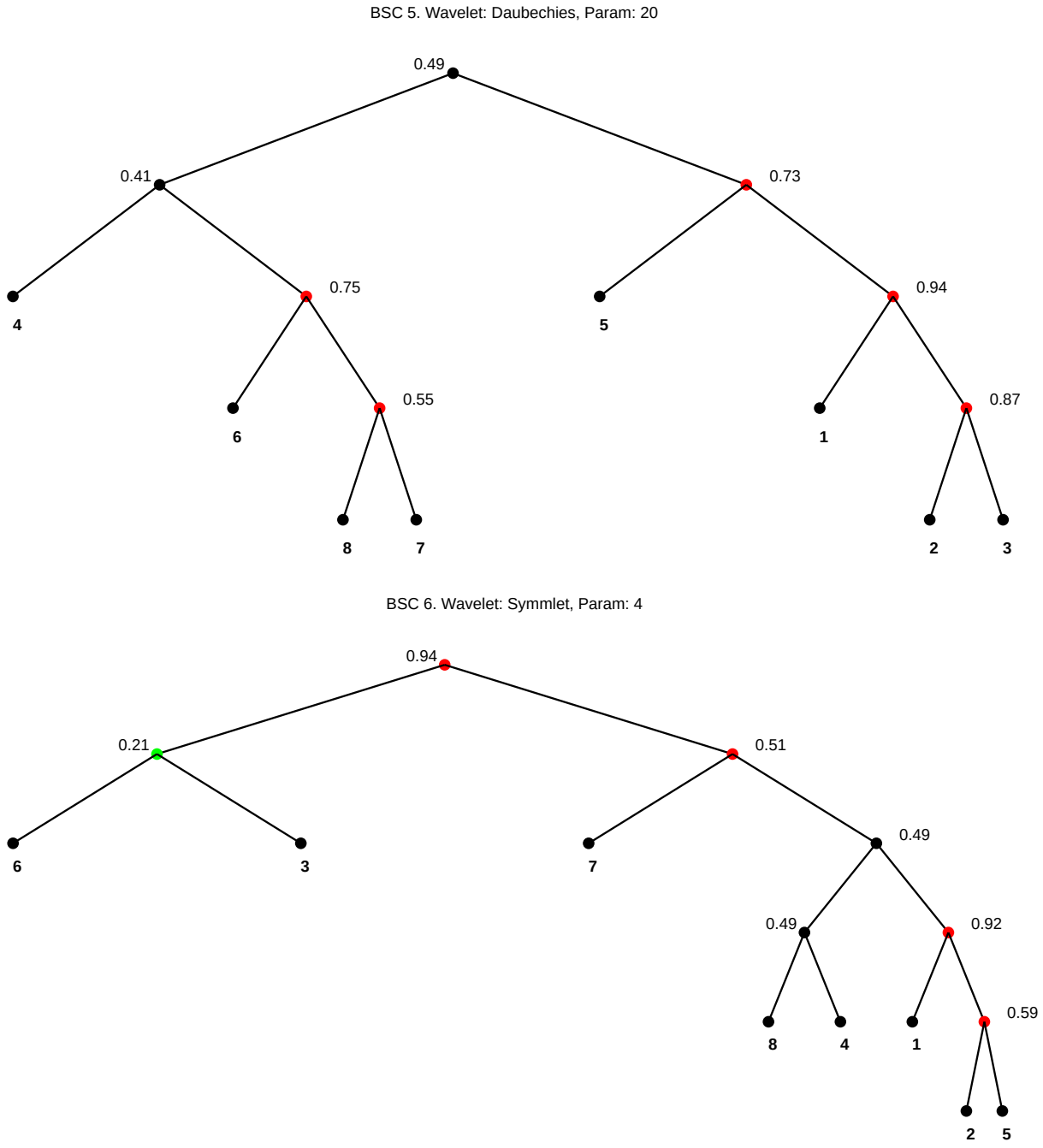


Figure 95: BSCs obtained for Experiment 2.2 (5–6 of 10).

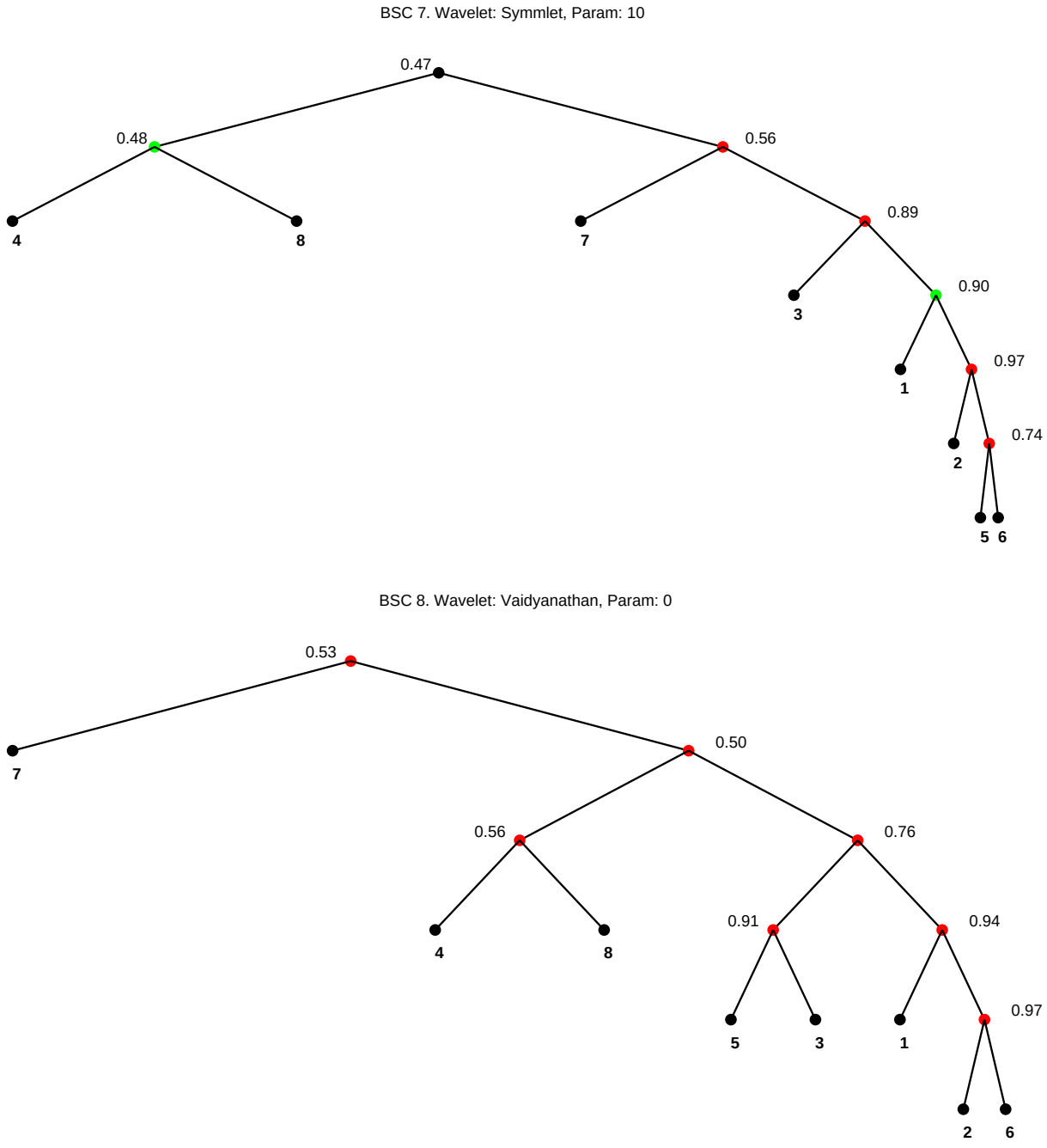


Figure 96: BSCs obtained for Experiment 2.2 (7–8 of 10).

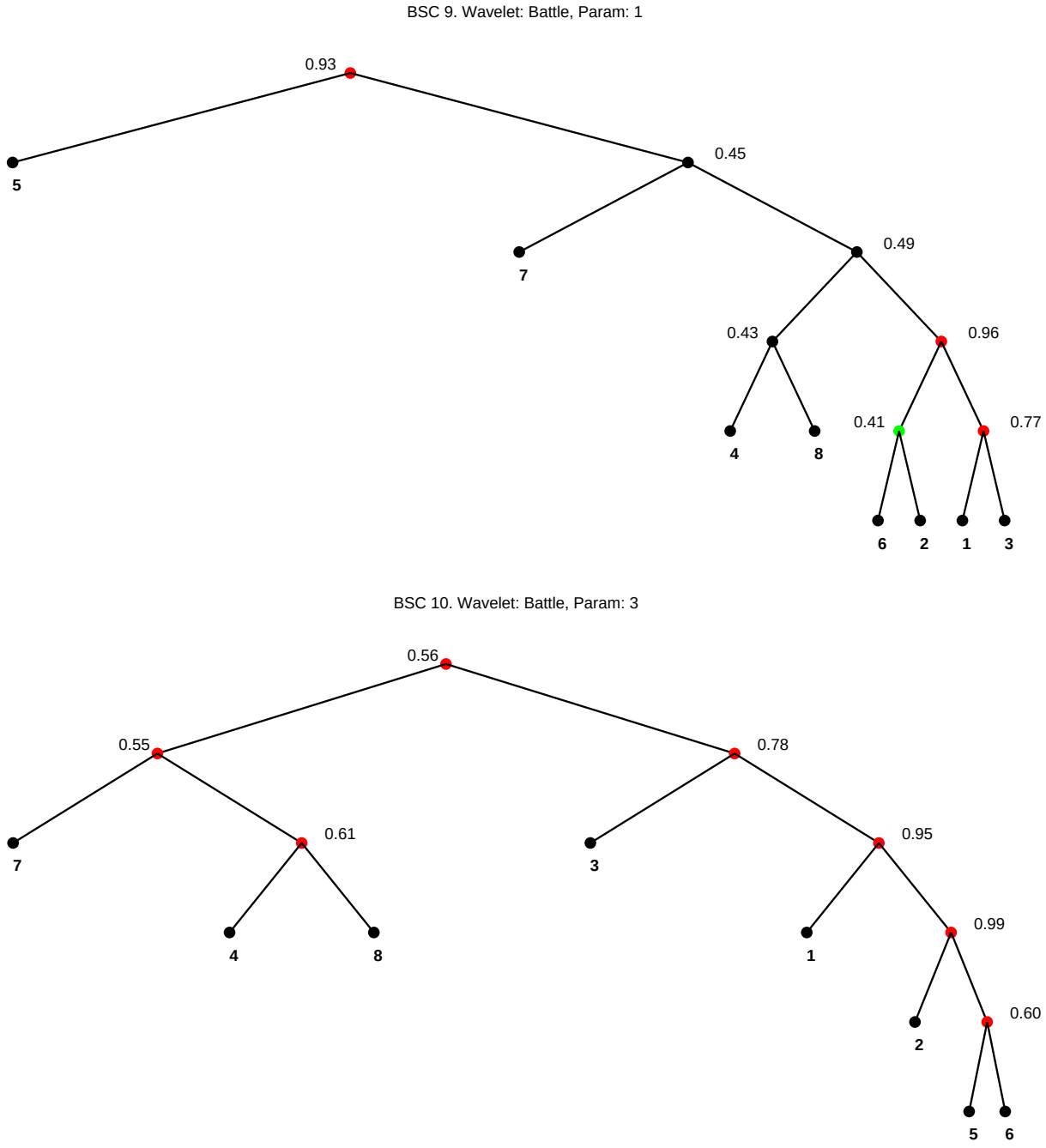


Figure 97: BSCs obtained for Experiment 2.2 (9–10 of 10).

Probability of Correct Classification The estimated probabilities of correct classification for Experiment 2.2 are shown in Figure 98. The forty BTCs are obtained as follows. The first ten BTCs are the basic BTCs, one per wavelet in Table 4. The remaining thirty BTCs are derived from these ten BTCs by keeping the structure, changing the CID, and retraining. The most obvious result is that the classifier performance is not particularly sensitive to the wavelet choice. Six of the BSCs achieve a P_{cc} of greater than 0.98; the rest are near 0.9. The hypertree classifier outperforms all forty of its constituent BTCs but is generally outperformed by the BSCs.

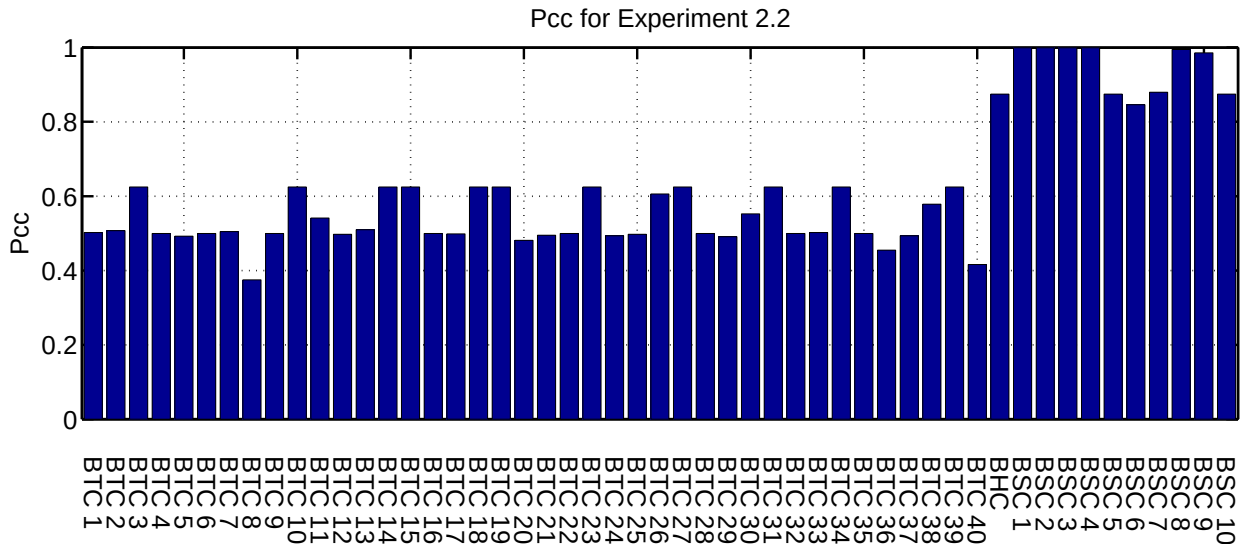


Figure 98: Probability of correct classification for Experiment 2.2.

Confusion Matrices and Quality Measures The confusion matrices and histograms of decision quality are not particularly informative and are not shown.

Conclusions for Experiment 2.2

1. The basic performance ordering, predicted by theory, of BSC $>$ BHC $>$ $\{$BTC$\}$ holds, as in Experiment 2.1.
2. Classifier performance is not strongly sensitive to the wavelet choice, unlike the one-dimensional experiment. This is likely due to the fact that the performance here is dominated by severe ambiguities which cannot be influenced by a better match between the classes and the wavelet shape.

5.2.3 Experiment 2.3: Path Correction in the BTC

In this final experiment, we revisit Experiment 2.1 and allow the BTCs and BSC to employ the path-correction algorithm. Recall that this optional algorithm greatly improved performance in Experiment 1. In Experiment 2.3, we employ the structures obtained in Experiment 2.1, but allow path correction to take place.

Probability of Correct Classification The estimated probability of correct classification for Experiment 2.3 is shown in Figure 99. By comparing with Figure 67, we see that the performances for the BTCs are slightly worse in Experiment 2.3, and the performances for the BHC and BSC are unchanged. So, path correction does not help here. It may have helped with the BSC, but its performance without path correction was already nearly perfect, so there are few paths to correct.

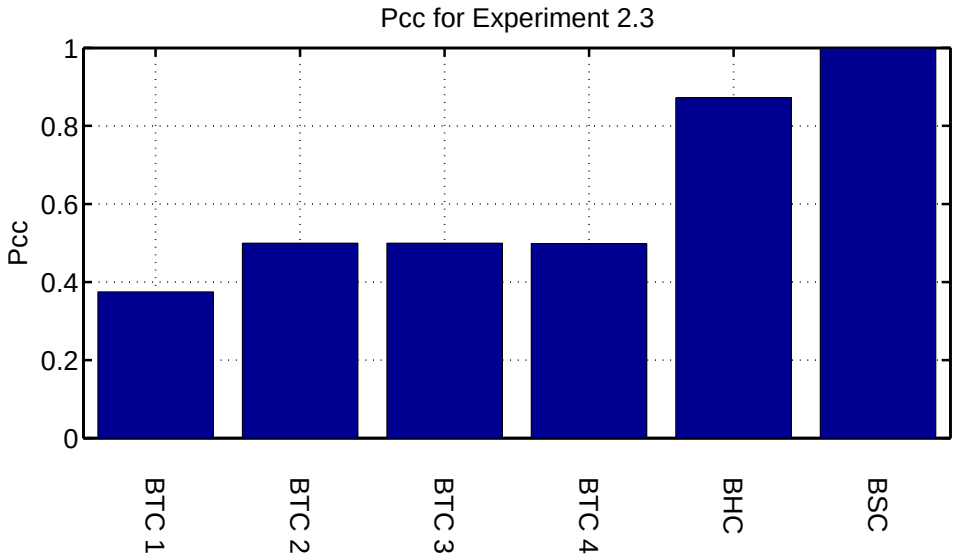


Figure 99: Probability of correct classification for Experiment 2.3.

Confusion Matrices The confusion matrices for Experiment 2.3 are shown in Figures 100–102.

Quality Measures The histograms of the output quality measure for Experiment 2.3 are shown in Figures 103–105. As in Experiment 2.1, there are many incorrect decisions with very high quality.

Conclusions for Experiment 2.3

1. Path correction does not help a BTC (or BSC) when the problem contains severe ambiguities. This is because such ambiguities result naturally in incorrect decisions with high quality. Even when the classifier attempts to correct a path, it will likely end up choosing one of the decisions in the equivalence class made up by the ambiguous classes.

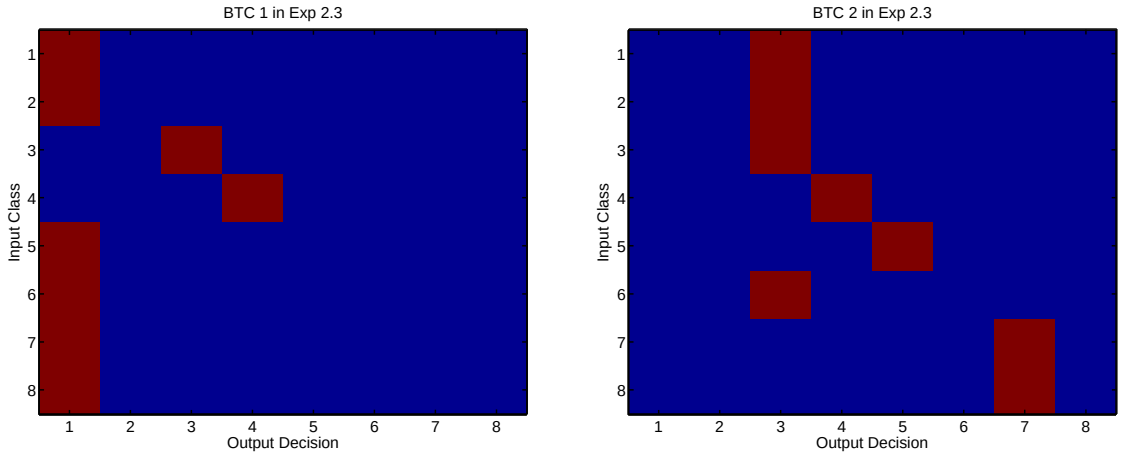


Figure 100: Confusion matrices for BTCs 1 and 2 in Experiment 2.3.

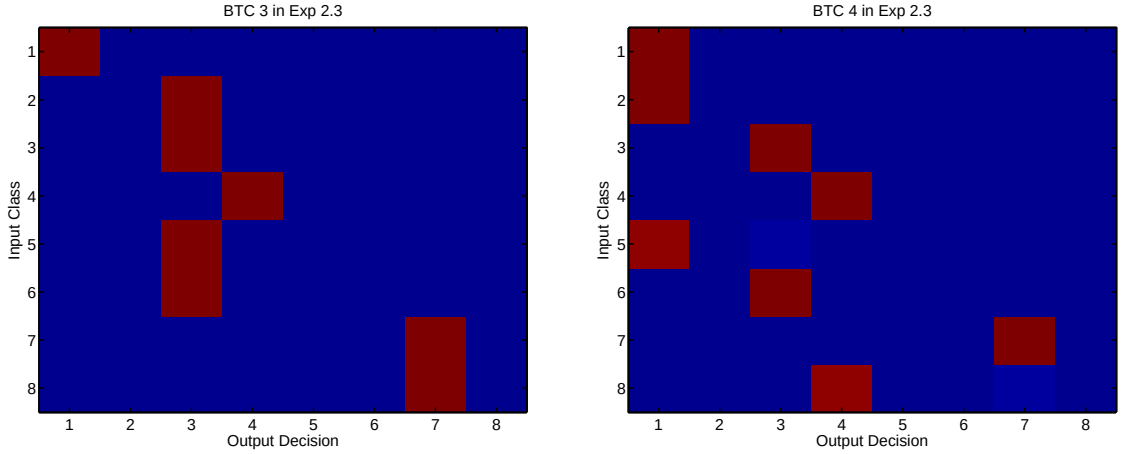


Figure 101: Confusion matrices for BTCs 3 and 4 in Experiment 2.3.

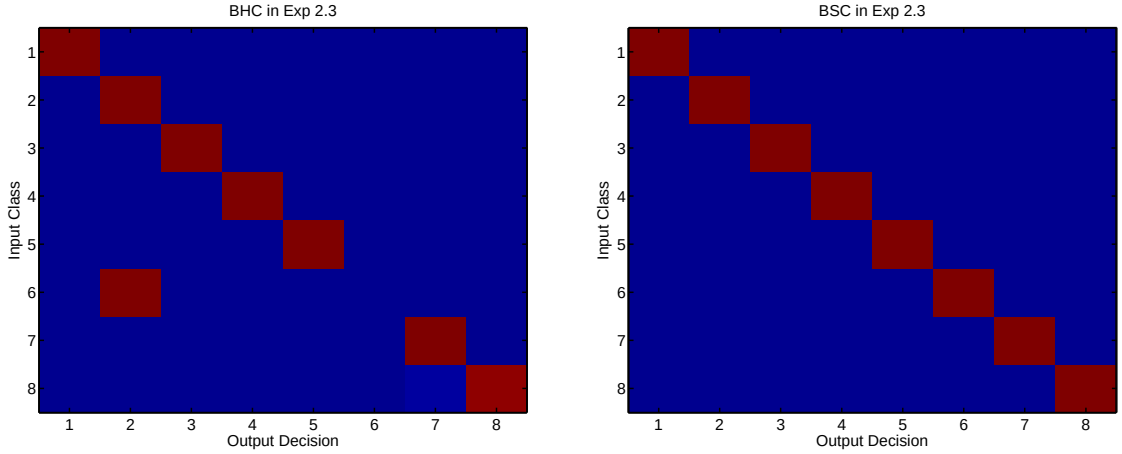


Figure 102: Confusion matrices for the BHC and BSC in Experiment 2.3.

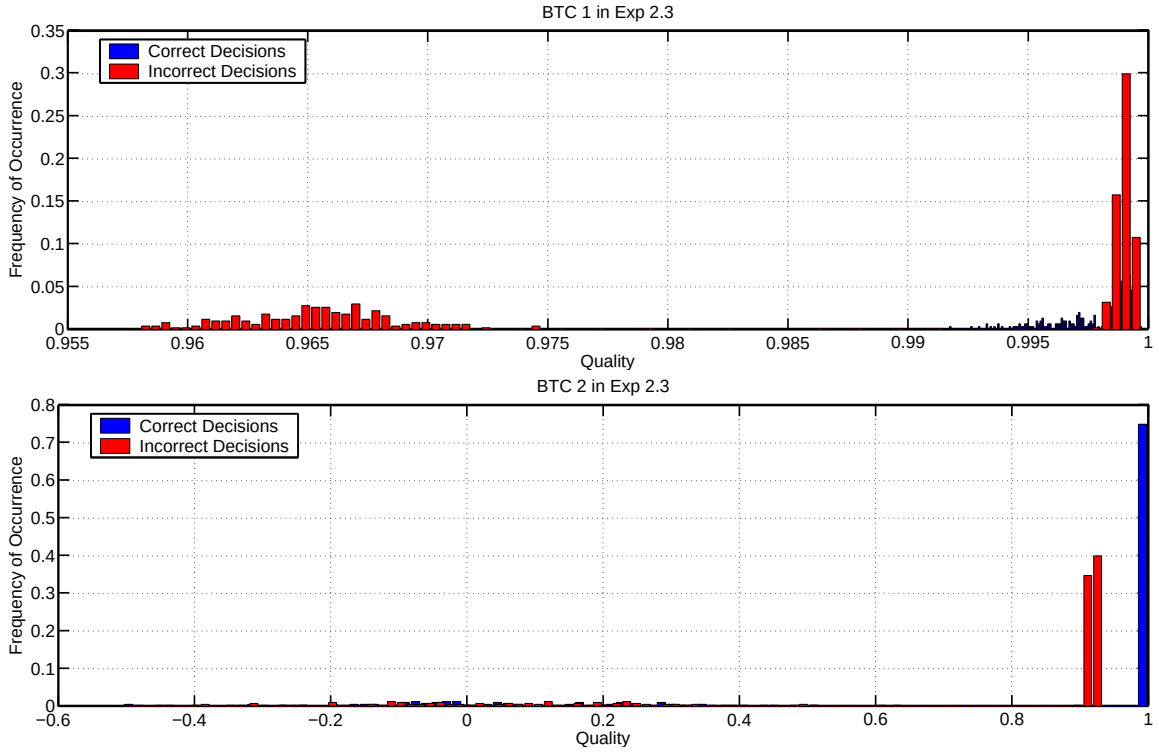


Figure 103: Quality histograms for BTCs 1 and 2 in Experiment 2.3.

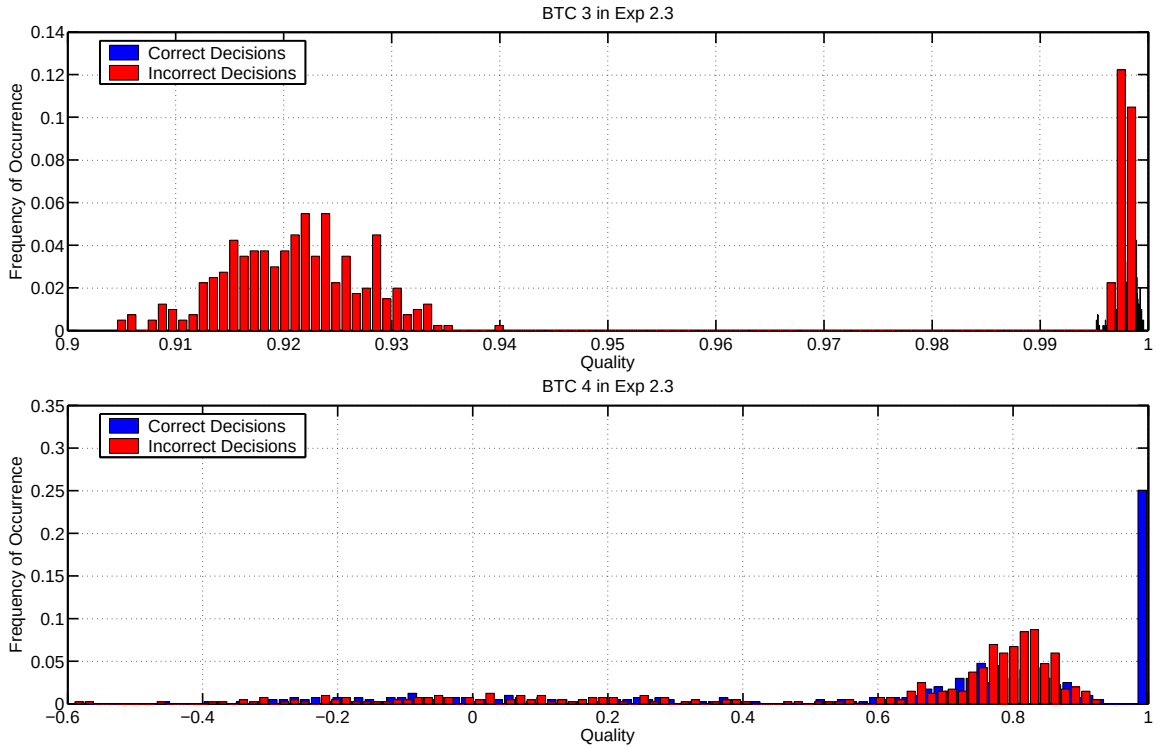


Figure 104: Quality histograms for BTCs 3 and 4 in Experiment 2.3.

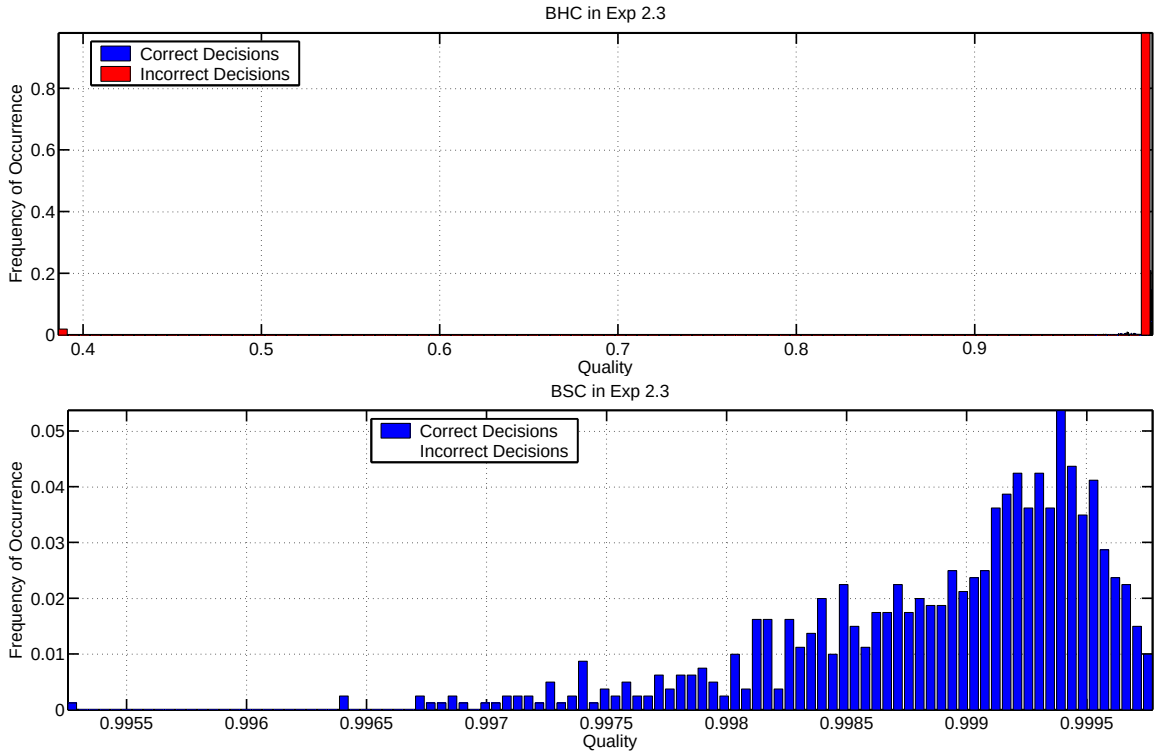


Figure 105: Quality histograms for the BHC and BSC in Experiment 2.3.

5.3 Collected-Data Problem: The StatLog Data Sets

This section presents the results of various experiments performed by applying the binary tree classifiers (BTCs) and the binary hypertree classifier (BHC) to publicly available data sets intended for use by classifier-algorithm developers.

In particular, we present test results for the StatLog data sets [9]. The StatLog data sets are publicly available data sets obtained from the Laboratory of Artificial Intelligence and Computer Science (LIACC) at University of Porto. These data sets were used in the "Project StatLog" of the Machine Learning subgroup for evaluation and characterization of machine learning, neural, and statistical classification algorithms. Some of the data sets in the StatLog database were originally obtained from the larger database at the UCI Machine Learning Repository [10]. The test results of this section aid in the evaluation of the overall BTC and BHC performance and robustness.

5.3.1 Data Set Description

There are a total of ten different data sets in the StatLog repository. Four of these data sets (DNA, Letter, Shuttle, Satimage), which already have the training and test data sets split, are used to obtain the results presented herein. Some of the data sets are claimed to have been "processed." For example, for the DNA sequence set, the data in the file is already converted to numerical values from the original symbolic variables representing the nucleotides. Table 5 tabulates some parameters pertaining to these four data sets.

A brief description of each of the four data sets follows.

1. **DNA:** Each entry in the data set represents a DNA sequence with splicing boundaries to be classified. The original 60 symbolic variables/attributes in the sequence representing the nucleotides have been converted into 180 binary indicator variables. Specifically, the four alphabet symbols (A, C, G, T) representing the nucleotides are mapped into the four binary sequences: 100, 010, 001, and 000 respectively. A note posted on the repository site states that much better performance is generally observed if attributes closest to the junction are used, and these correspond to attribute indices 61-120.

2. **Letter:** The original black-and-white pixel displays of the 26 capital letters in 20 different fonts randomly distorted are converted into 16 primitive numerical attributes representing statistical moments and edge counts and then scaled to the range of integers from 0 to 15 to form this data set. This is the only preprocessing done to the data set.

3. **Shuttle:** No explicit data description is available for this data set. The class labels appear to indicate some sort of shuttle control signals, such as Rad Flow, Fpv Close, Fpv Open, Bypass, etc. The only preprocessing done to the data set was the stripping away of the time ordering information by randomizing the order in which the original data vectors came. This ordering information could be relevant in classification that takes into account control sequencing information.

4. **Satimage:** This data set represents data from Landsat satellite images. Each sample represents a 3-by-3 square image within an 82-by-100 pixel area that is contained in the original image of 2340-by-3380 pixels. The images correspond to digital images of the same scene in 4 different spectral bands. Class labels include red soil, cotton crop, grey soil etc. Each

data vector comprises the range of values (from 0 for black to 255 for white) of the 9 pixels in 4 different spectral bands resulting in 36 attributes. A note in the data description states that to avoid the problem which arises when a 3-by-3 neighborhood straddles a boundary, one can consider using only the 4 attributes 17-20 which correspond to the 4 spectral values for the center pixel. No preprocessing was done to this data set.

5.3.2 Experimental Set-Up

A total of 16 experiments were performed on the StatLog data sets by varying a number of parameters including the data set itself, the number of input vectors used to train the BTC, wavelet types, data dimension (the number of attributes used), processed data dimension (dimension after any zero padding), feature length K, and wavelet tree depth J. Relevant parameters for each individual experiment are summarized in Tables 8-11. The parameters that are varied in the 16 experiments are tabulated in Table 6. In these tables, N_{tr} and N_{te} denote, respectively, the number of training and test input vectors used in the experiments. In order to expedite processing, the first N_{tr} data vectors in each class of the training data sets were used to train the BTC, and the resulting classifier structures were then applied to the first N_{te} data vectors of the test data sets. It is noted, however, that the data distribution is not uniform over the data class. That is, the number of data vectors available for each class is different. For the Shuttle data set, about 80% of the data belongs to Class 1, and the distribution ranges from 6 to 34108 for the training set and from 2 to 11478 for the test set. For the remaining three data sets, there are at least 100 vectors per class. Therefore, the number of training and test data vectors used in the experiment for the Shuttle data is chosen to be ≤ 100 (i.e. utilizing all vectors given in a class whose size is smaller than 100). For the remaining data sets, the number of test data vectors used is chosen to be $N_{te} = 100$. The number N_{tr} of training vectors used for each data set is shown in Table 6.

To further expedite processing, only the first 8 classes (letters A-H) were used for the Letter data set experiment while two different numbers of training vectors were used for comparison purposes in this experiment. Three different wavelet types were used for the Letter and Satimage data sets in order to observe any sensitivity to the choice of wavelets. Data dimension is varied for only the DNA and Satimage data sets according to the notes given in Section 5.3.1 on the possible effects of the choice of attributes on performance. The processed data dimension is simply the dimension of the data vector zero padded to fulfill a dyadic dimension restriction. Values of K and J are also varied to study their effects on performance. It is noted however that these 16 experiments are not meant to be exhaustive. They simply serve the purpose of providing insight into the applicability of the BTC in classifying inputs of various types.

5.3.3 Results and Discussion

Results of the 16 experiments are shown in Figures 106-153. These correspond to 16 sets of three figures, one for each experiment. Results of experiments with path correction are also shown in each figure for comparison purpose. The following results are presented in the set of three figures:

1. Quality measures as a function of trial indices. The trial indices are ordered in groups, one for each class. The quality measure is the quality of the decision made by traversing the classification tree. This quality is a function of the set of correlation coefficient pairs

encountered during the tree traversal. Also indicated are the probability of error P_e and probability of correction classification P_{cc} over all classes and all trials.

2. Confusion matrices that show the basic misclassification patterns.

3. Histograms of the quality measures corresponding to trials for which the classification is correct (indicated by red bars) and trials for which the classification is incorrect (indicated by blue bars).

Table 7 summarizes the individual (single class) and overall classification performance in terms of P_{cc}.

Based on the figures, there does not seem to be noticeable overall improvement with the use of path correction in these data sets. While improvement is observed in the results for certain classes and data sets, degradation is observed in others. This is possibly due to the nature of the data such as interclass ambiguity. The effect of path correction is therefore not conclusive from these experiments.

For the DNA data set, an overall performance degradation is observed in Experiment 2 with the use of the full data dimension versus the use of only attributes 61-120 in Experiment 1, which is claimed to yield better results (cf. DNA data set description in Section 5.3.1). However, it is noted that while performance is worsened for Classes 1 and 3, performance for Class 2 is improved.

For the Letter data set, the use of a larger number of training vectors (500 instead of 100) does not appear to improve performance noticeably between Experiments 3 and 4 for which $K = 8$. But for $K = 16$ (Experiments 5-10), the use of a larger number of training vectors appears to improve overall performance. Increasing K from 8 to 16 also appears to improve general performance slightly. It is interesting to note that with the larger K, performance for Classes 1-4 is improved while performance for Classes 5-8 is worse. It is also noted that the low-quality measures evident in the results for $K = 8$ are eliminated in the results for $K = 16$. Lastly, results of Experiments 5-10 for the Letter data set also demonstrate insensitivity to the particular choices of wavelets used.

For the Shuttle data set, the increase in K does not appear to affect performance significantly. Performance for certain classes is improved while others worsened.

Finally, for the Satimage data set, a slight performance degradation is observed in Experiment 14 with the use of the full data dimension versus the use of only attributes 17-20 in Experiment 13 (cf. Satimage data set description in Section 5.3.1). Also noticed is a small degree of sensitivity to the choice of wavelet types.

The experimental results presented here demonstrate that the BTC has the capability of classifying suitable publicly available data sets. While performance for certain classes and data sets are very good, performance for some others are quite bad. This can possibly be attributed to the nature of the data sets such as interclass ambiguity. The below-average performance for the Letter data set could possibly be due to the preprocessing that was done to the original letter image data to which we have not found access. Performance is believed to improve if the LDB-based classifier is applied directly to these original images rather than the numerical attributes of edge counts and statistical moments represented in these data.

Data Set	Size of Training Set	Size of Test Set	Number of Classes	Data Dimension	Integer Data Values
DNA	2000	1186	3	180	$\{0,1\}$
Letter	15000	5000	26	16	[0,15]
Shuttle	43500	14500	7	9	[-26739,15164]
Satimage	4435	2000	6	36	[0,255]

Table 5: Some parameters pertaining to four of the data sets obtained from the StatLog repository.

Experiment	Data Set	N_{tr}	Wavelet	Data Dimension	Processed Data Dimension	K	J
1	DNA	400	Coiflet(5)	60	64	16	3
2	DNA	400	Coiflet(5)	180	256	32	8
3	Letter	100	Coiflet(5)	16	16	8	4
4	Letter	500	Coiflet(5)	16	16	8	4
5	Letter	100	Coiflet(5)	16	16	16	4
6	Letter	100	Beylkin(0)	16	16	16	4
7	Letter	100	Daubechies(4)	16	16	16	4
8	Letter	500	Coiflet(5)	16	16	16	4
9	Letter	500	Beylkin(0)	16	16	16	4
10	Letter	500	Daubechies(4)	16	16	16	4
11	Shuttle	≤ 100	Coiflet(5)	9	16	4	4
12	Shuttle	≤ 100	Coiflet(5)	9	16	9	4
13	Satimage	400	Coiflet(5)	4	8	4	4
14	Satimage	400	Coiflet(5)	36	64	36	6
15	Satimage	400	Beylkin(0)	36	64	36	6
16	Satimage	400	Daubechies(4)	36	64	36	6

Table 6: Summary of varied parameters in the StatLog data experiments.

Exp	Data Set	N_{tr}	Wavelet	Data Dimension	K	J	$P_{cc_{min}}$	$P_{cc_{max}}$	P_{cc}
1	DNA	400	Coiflet(5)	60	16	3	0.71	0.95	0.82
2	DNA	400	Coiflet(5)	180	32	8	0.56	0.74	0.62
3	Letter	100	Coiflet(5)	16	8	4	0.33	0.93	0.51
4	Letter	500	Coiflet(5)	16	8	4	0.26	0.93	0.51
5	Letter	100	Coiflet(5)	16	16	4	0.18	0.90	0.53
6	Letter	100	Beylkin(0)	16	16	4	0.18	0.90	0.53
7	Letter	100	Daubechies(4)	16	16	4	0.18	0.90	0.53
8	Letter	500	Coiflet(5)	16	16	4	0.27	0.90	0.58
9	Letter	500	Beylkin(0)	16	16	4	0.27	0.90	0.58
10	Letter	500	Daubechies(4)	16	16	4	0.27	0.90	0.58
11	Shuttle	≤ 100	Coiflet(5)	9	4	4	0.28	1.00	0.69
12	Shuttle	≤ 100	Coiflet(5)	9	9	4	0.22	1.00	0.70
13	Satimage	400	Coiflet(5)	4	4	4	0.29	0.98	0.71
14	Satimage	400	Coiflet(5)	36	36	6	0.32	1.00	0.68
15	Satimage	400	Beylkin(0)	36	36	6	0.28	1.00	0.71
16	Satimage	400	Daubechies(4)	36	36	6	0.33	0.96	0.65

Table 7: Performance summary for the case with no path correction. $P_{cc_{min}}$ represents the probability of correct classification for the class that is classified correctly the least among the classes in the set and $P_{cc_{max}}$ represents the probability of correct classification for the class that is classified correctly the most. P_{cc} is the overall probability of correct classification.

Parameter	Exp1	Exp 2
Wavelet Type	Coiflet	Coiflet
Wavelet Parameter	5	5
Feature Length K	16	32
Number of Classes C	3	3
BTC Wavelet Tree Depth J	3	8
Data Dimension	[1 60]	[1 180]
Processed Data Dimension	[1 64]	[1 256]
N_{tr}	400	400
N_{te}	100	100

Table 8: Parameters for Experiments 1 and 2: DNA data set.

Parameter	Exp 3	4	5	6	7	8	9	10
Wavelet Type	Coiflet	Coiflet	Coiflet	Beylkin	Daub.	Coiflet	Beylkin	Daub.
Wavelet Parameter	5	5	5	0	4	5	0	4
Feature Length K	8	8	16	16	16	16	16	16
Number of Classes C	8	8	8	8	8	8	8	8
Wavelet Tree Depth J	4	4	4	4	4	4	4	4
N_{tr}	100	500	100	100	100	500	500	500
N_{te}	100	100	100	100	100	100	100	100

Table 9: Experimental parameters for Experiments 3–10: Letter data set. The data dimension and processed data dimensions are [1 16].

Parameter	Exp 11	Exp 12
Wavelet Type	Coiflet	Coiflet
Wavelet Parameter	5	5
Feature Length K	4	9
Number of Classes C	7	7
BTC Wavelet Tree Depth J	4	4
Data Dimension	[1 9]	[1 9]
Processed Data Dimension	[1 16]	[1 16]
N_{tr}	≤ 100	≤ 100
N_{te}	≤ 100	≤ 100

Table 10: Experimental parameters for Experiments 11 and 12: Shuttle data set.

Parameter	Exp 13	14	15	16
Wavelet Type	Coiflet	Coiflet	Beylkin	Daub.
Wavelet Parameter	5	5	0	4
Feature Length K	4	36	36	36
Number of Classes C	6	6	6	6
BTC Wavelet Tree Depth J	4	6	6	6
Data Dimension	[1 4]	[1 36]	[1 36]	[1 36]
Processed Data Dimension	[1 8]	[1 64]	[1 64]	[1 64]
N_{tr}	400	400	400	400
N_{te}	100	100	100	100

Table 11: Experimental parameters for Experiments 13–16: Satimage data set.

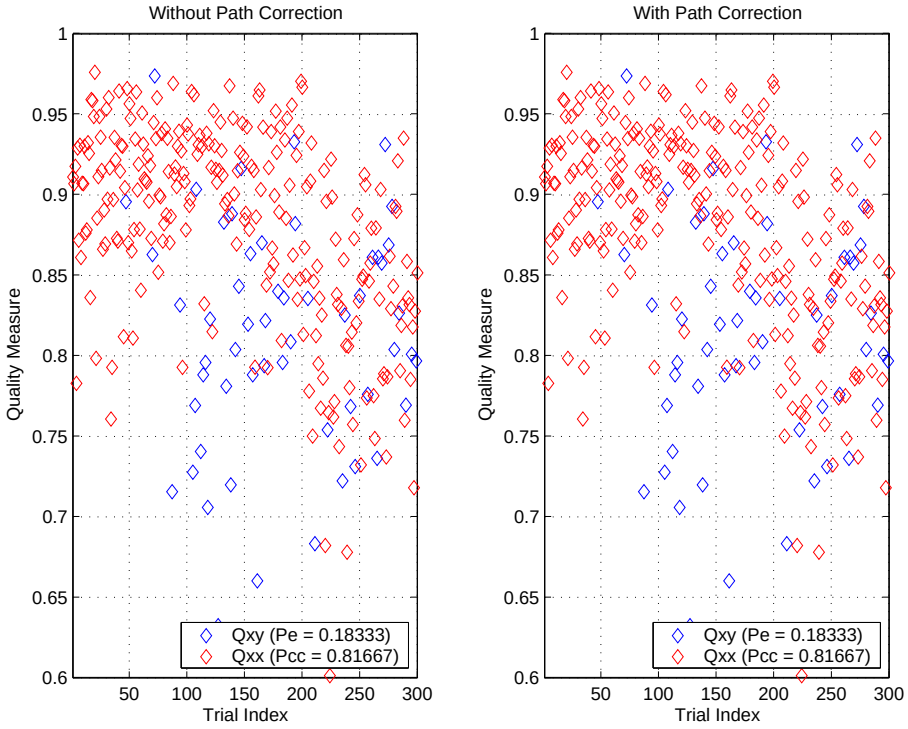


Figure 106: Quality measures for Experiment 1–DNA data set.

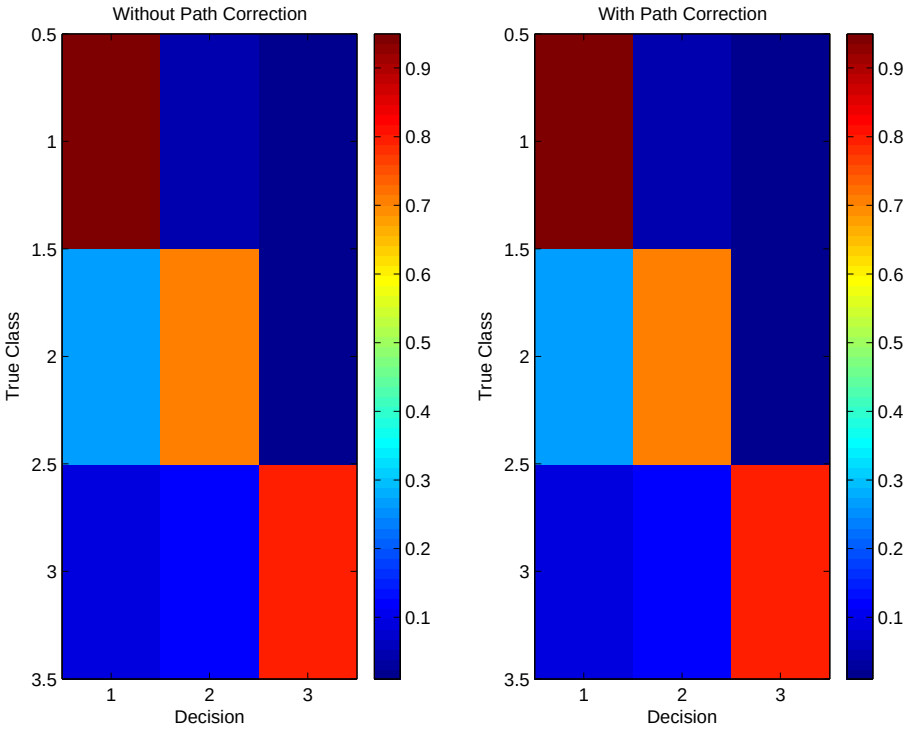


Figure 107: Confusion matrix for Experiment 1–DNA data set.

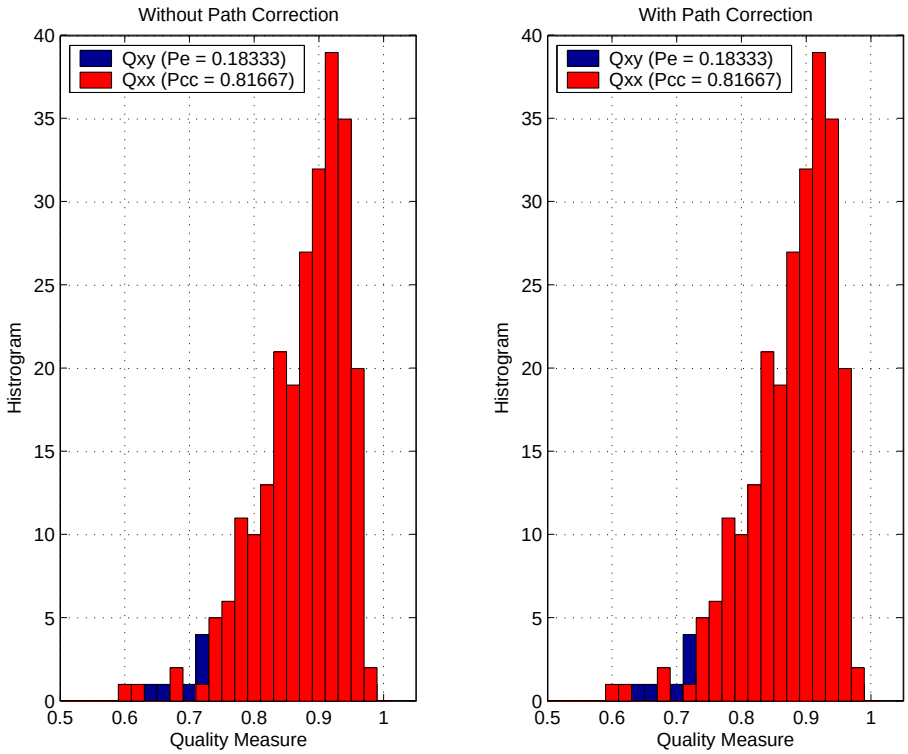


Figure 108: Histogram of quality measures for Experiment 1–DNA data set.

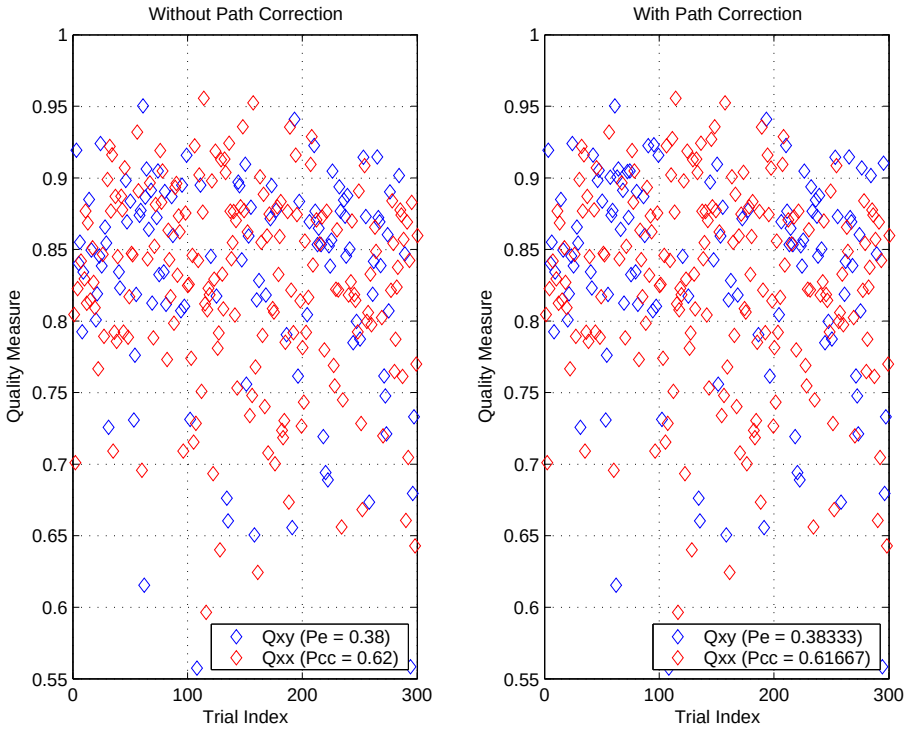


Figure 109: Quality measures for Experiment 2–DNA data set.

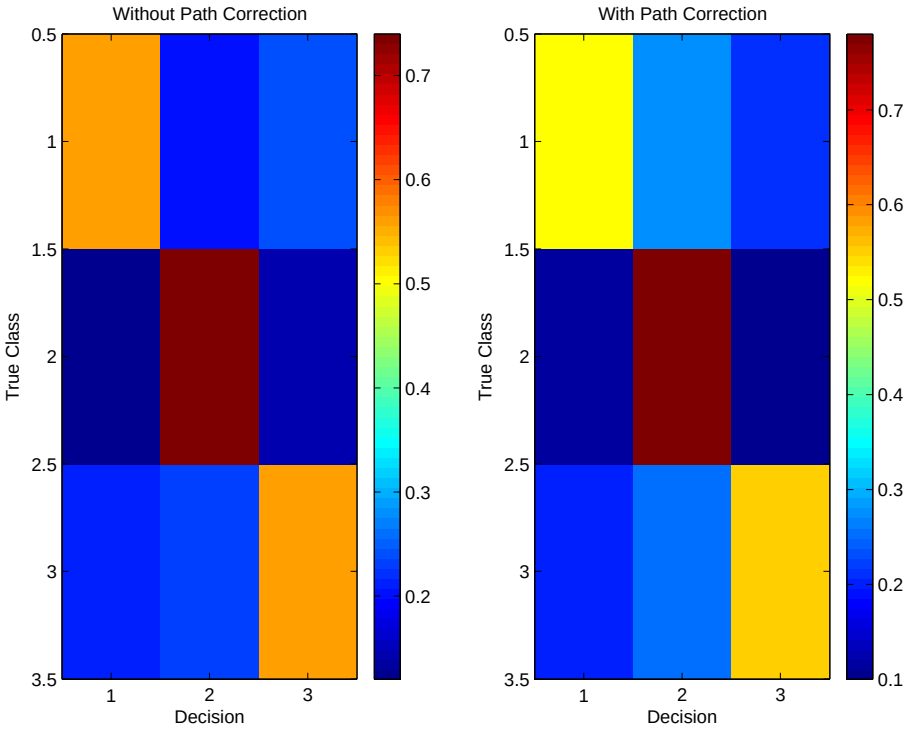


Figure 110: Confusion matrix for Experiment 2–DNA data set.

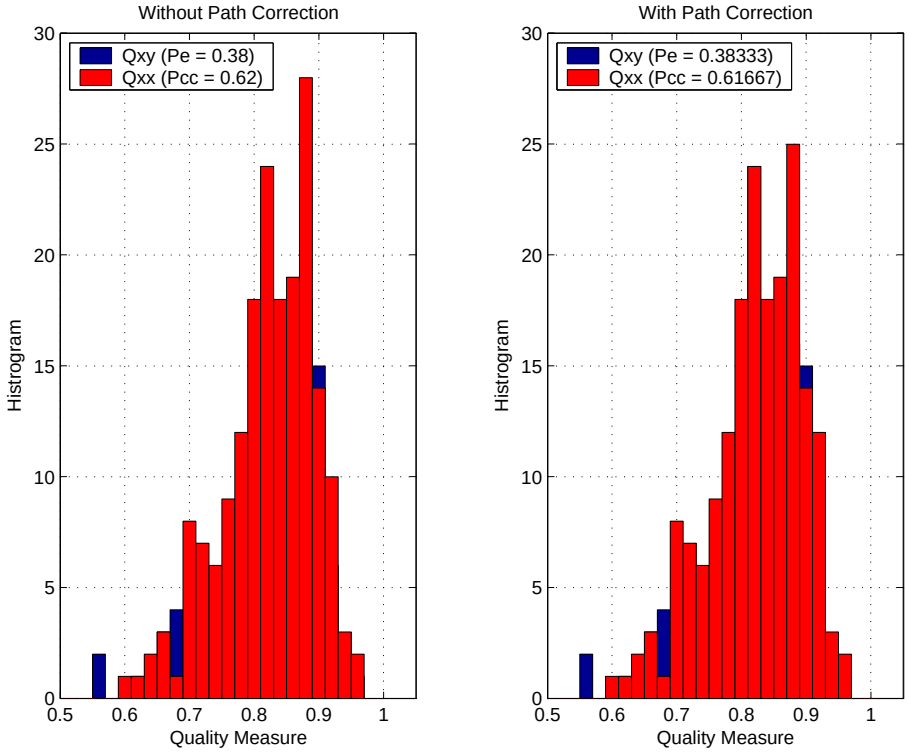


Figure 111: Histogram of quality measures for Experiment 2–DNA data set.

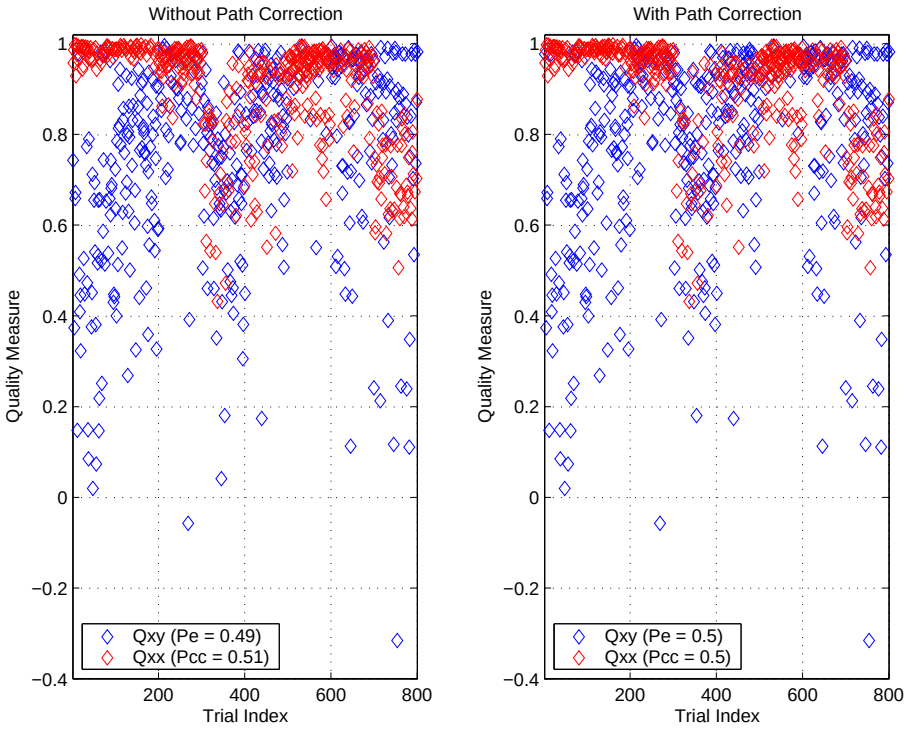


Figure 112: Quality measures for Experiment 3–Letter data set.

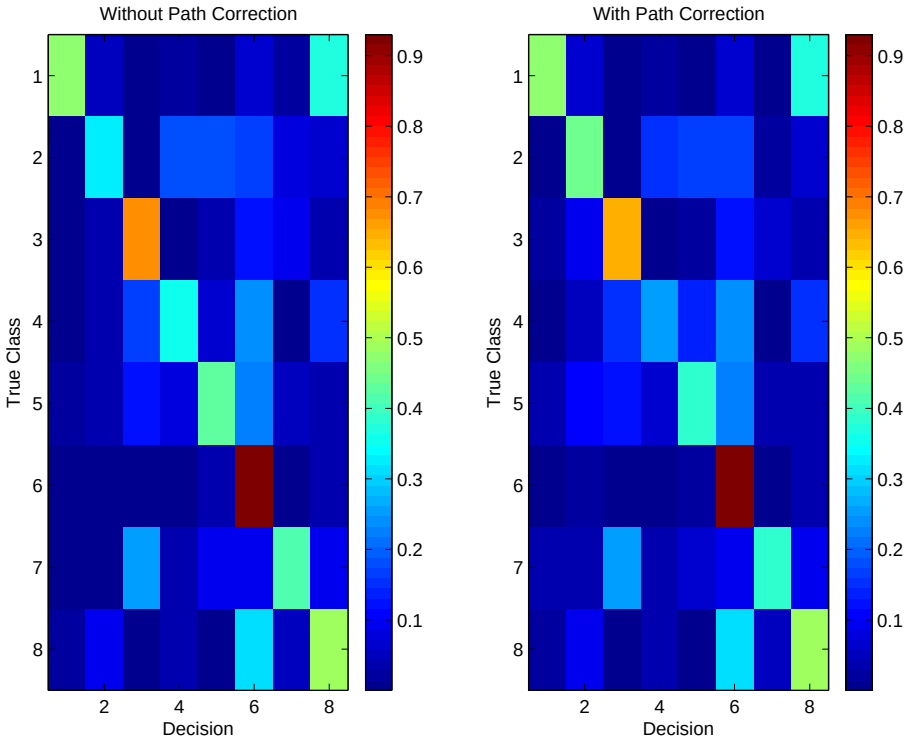


Figure 113: Confusion matrix for Experiment 3–Letter data set.

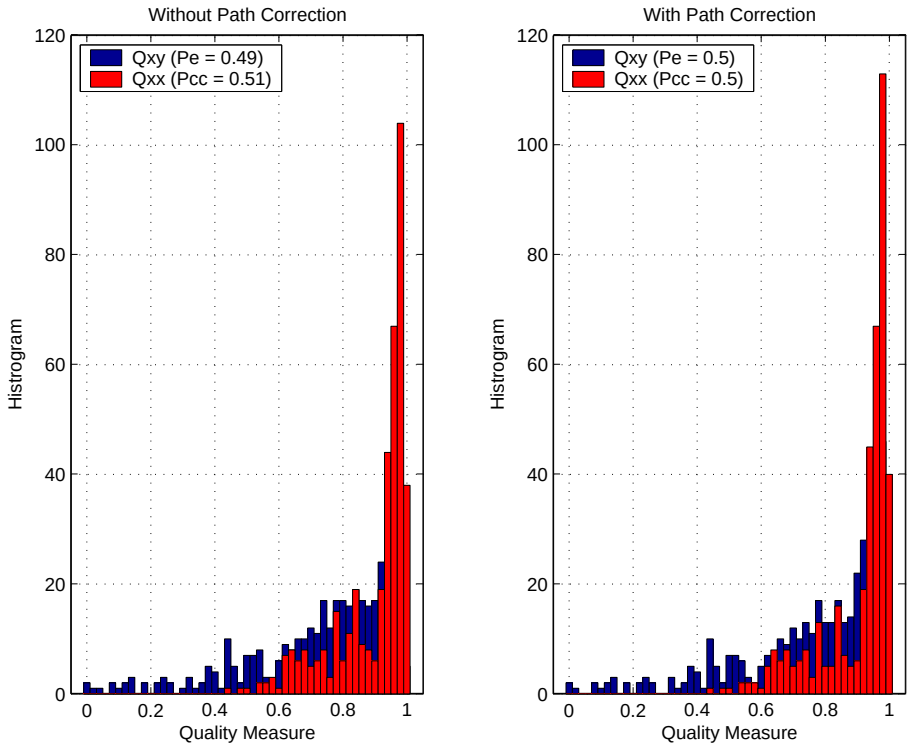


Figure 114: Histogram of quality measures for Experiment 3–Letter data set.

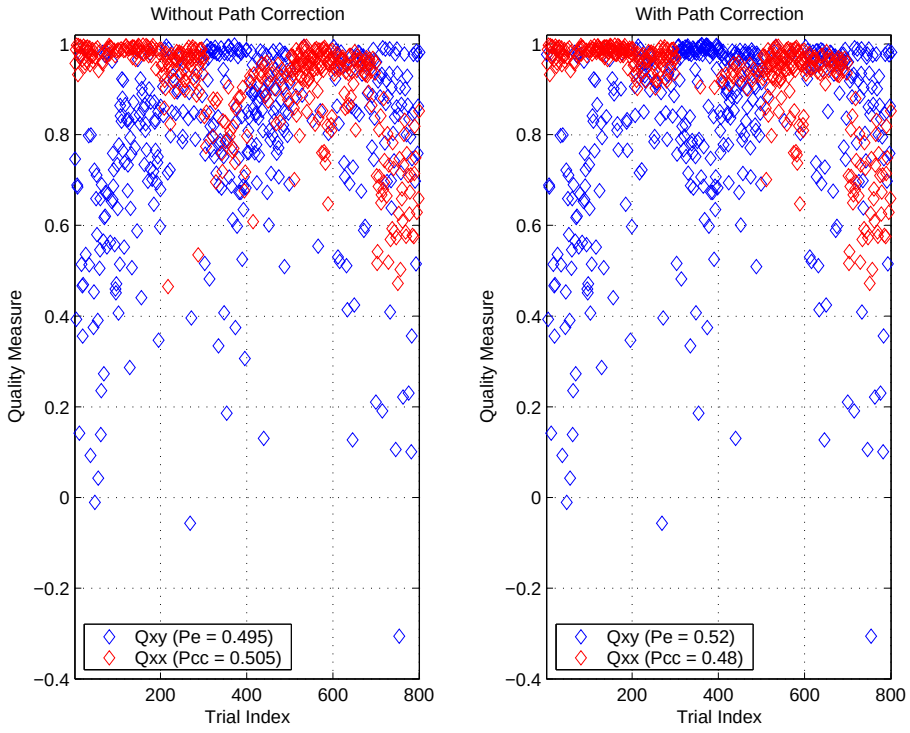


Figure 115: Quality measures for Experiment 4–Letter data set.

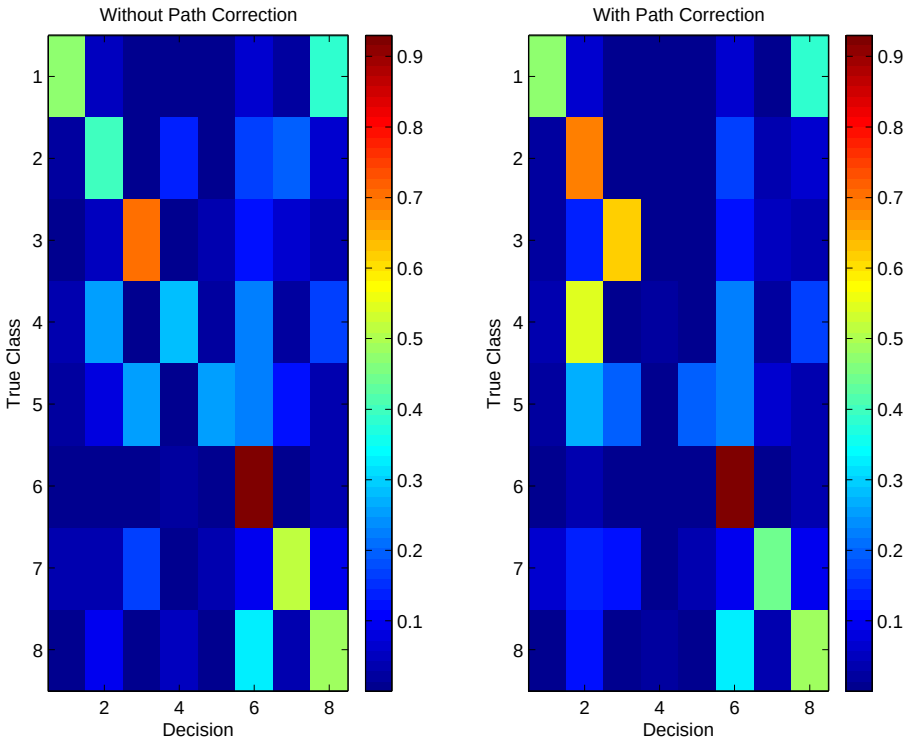


Figure 116: Confusion matrix for Experiment 4–Letter data set.

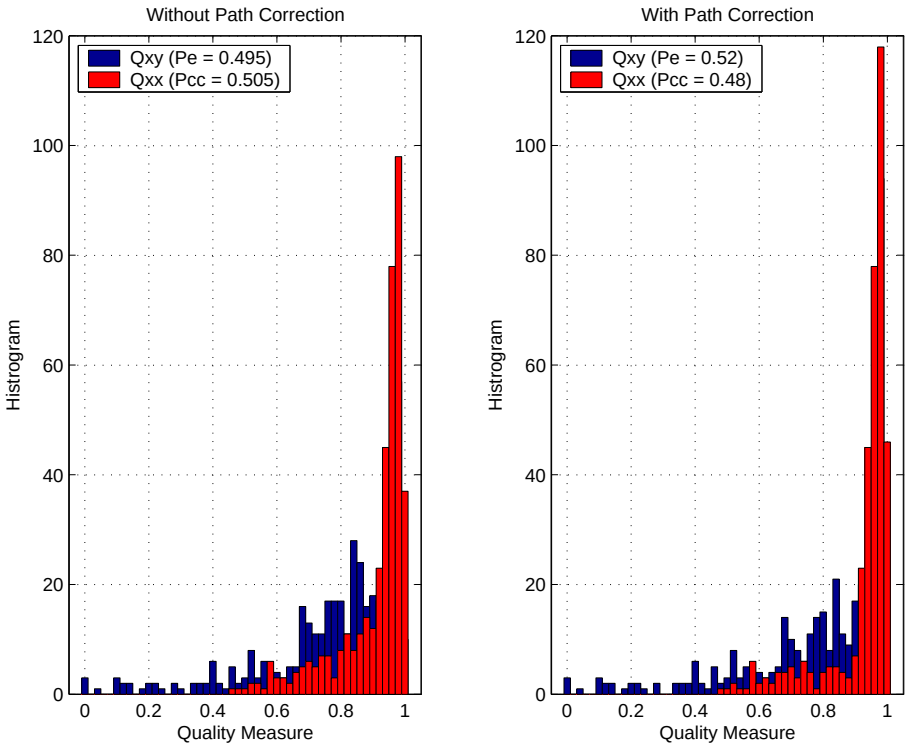


Figure 117: Histogram of quality measures for Experiment 4–Letter data set.

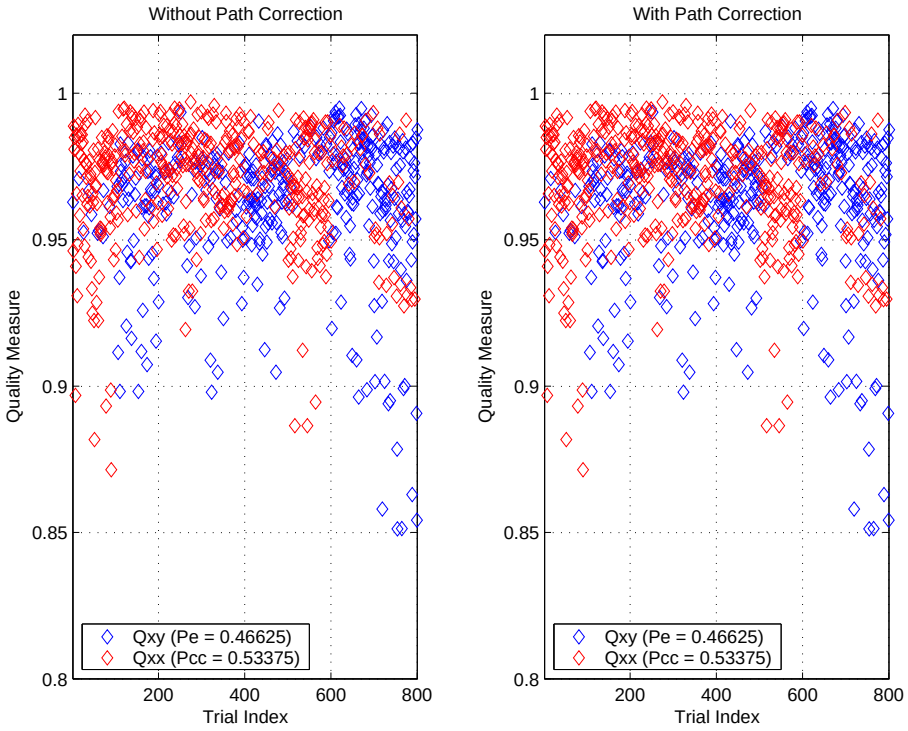


Figure 118: Quality measures for Experiment 5–Letter data set.

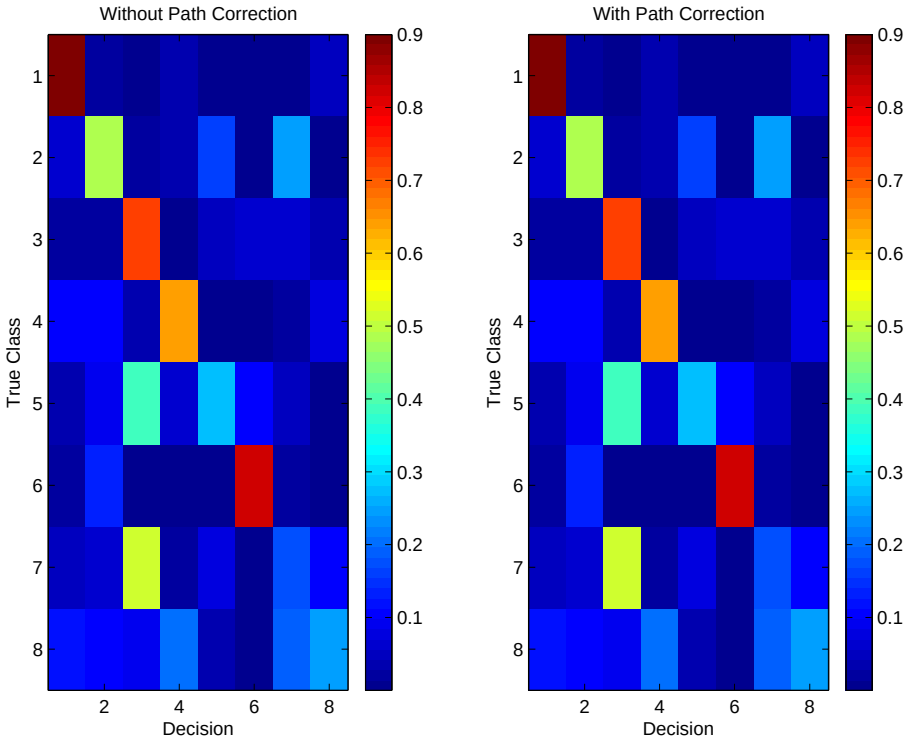


Figure 119: Confusion matrix for Experiment 5–Letter data set.

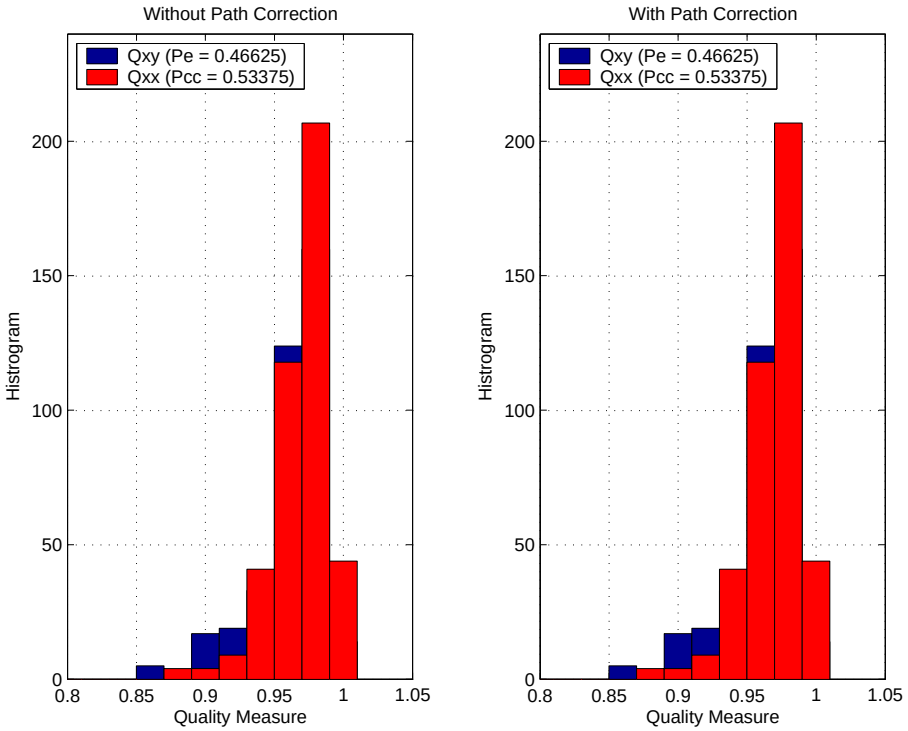


Figure 120: Histogram of quality measures for Experiment 5–Letter data set.

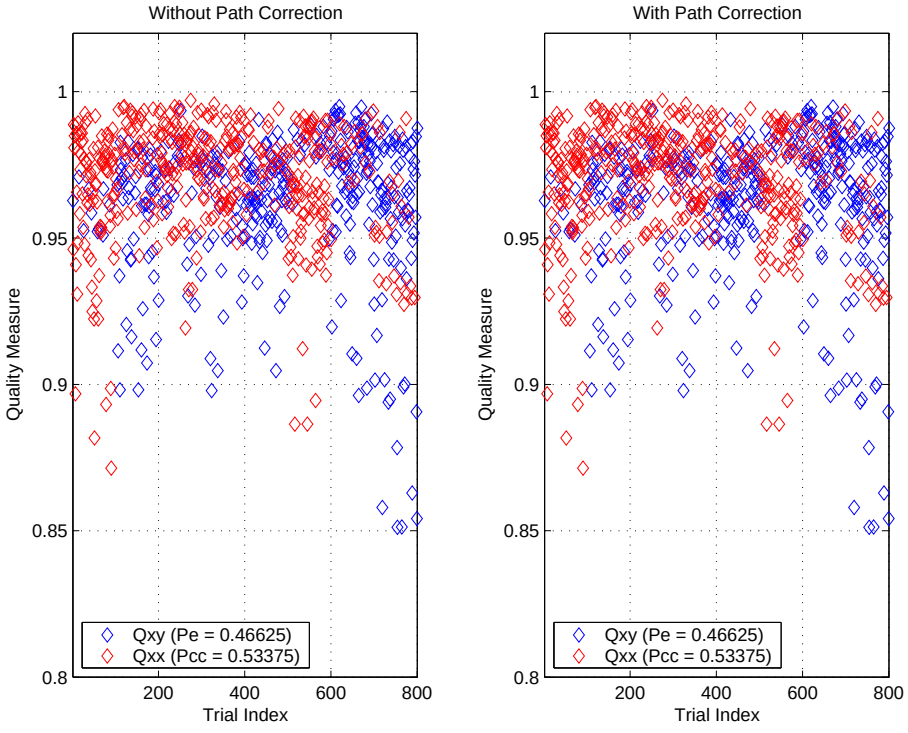


Figure 121: Quality measures for Experiment 6–Letter data set.

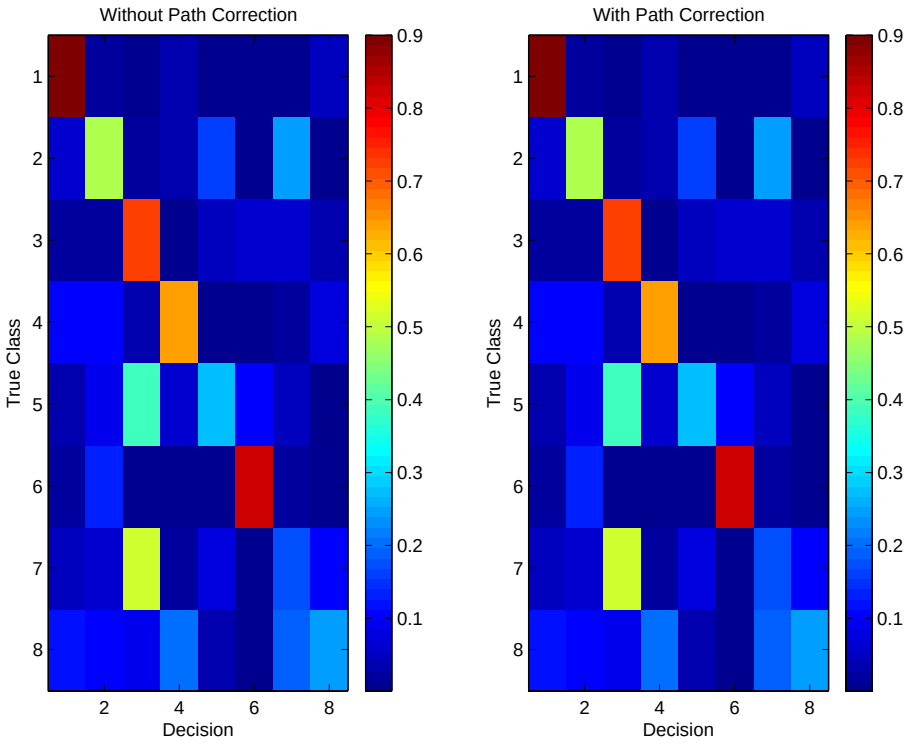


Figure 122: Confusion matrix for Experiment 6–Letter data set.

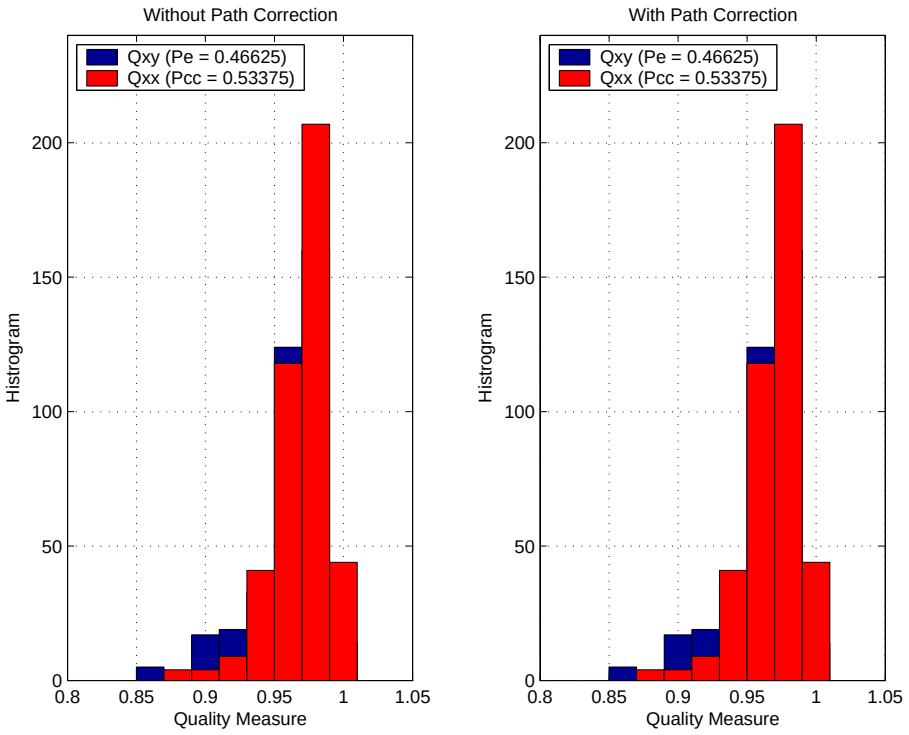


Figure 123: Histogram of quality measures for Experiment 6–Letter data set.

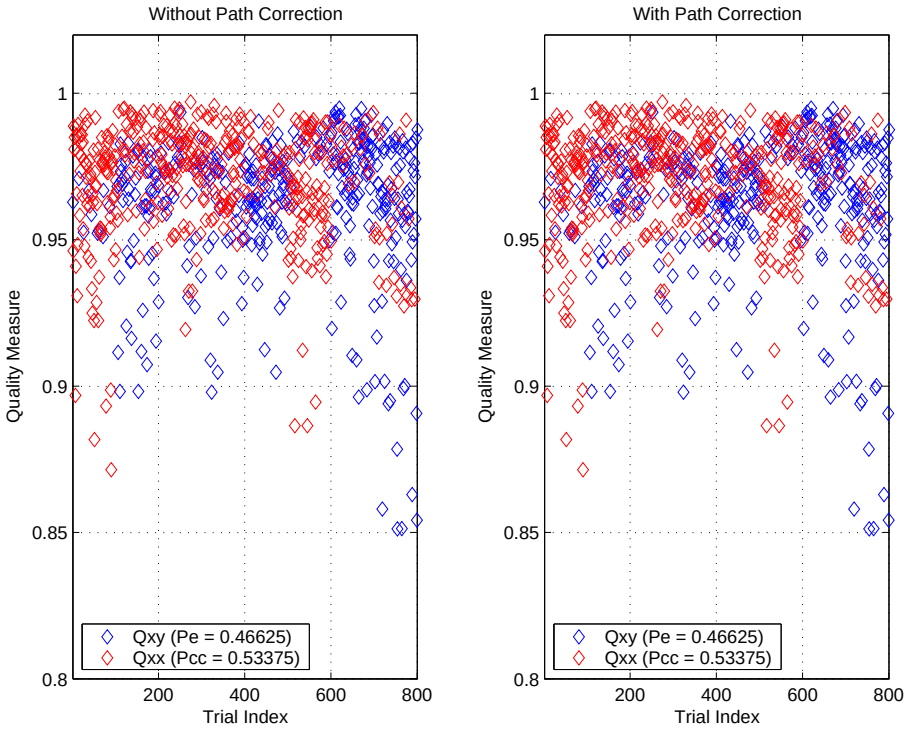


Figure 124: Quality measures for Experiment 7–Letter data set.

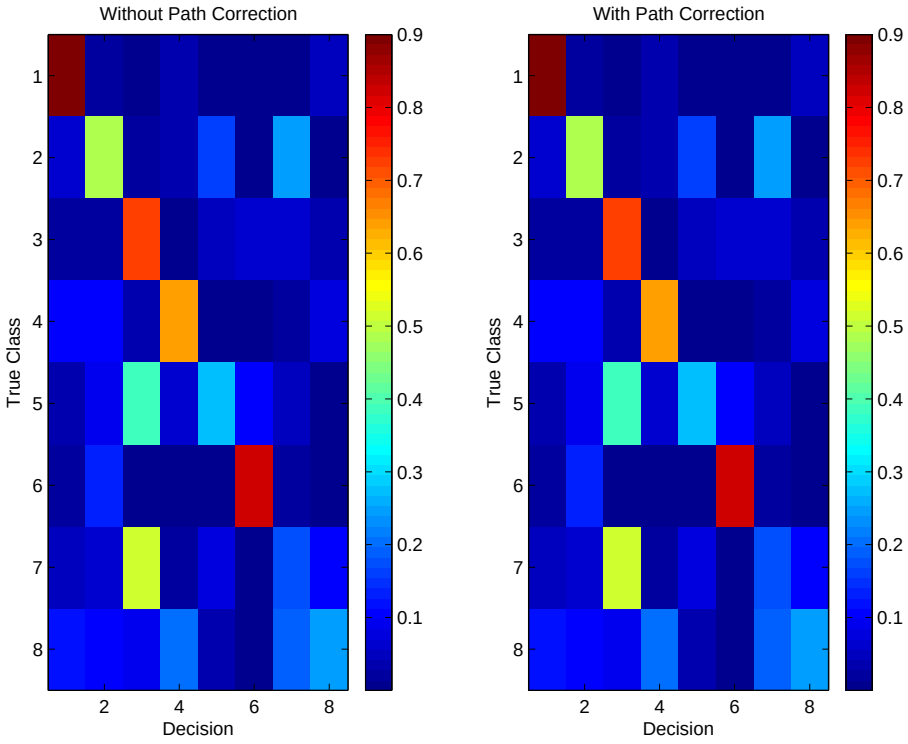


Figure 125: Confusion matrix for Experiment 7–Letter data set.

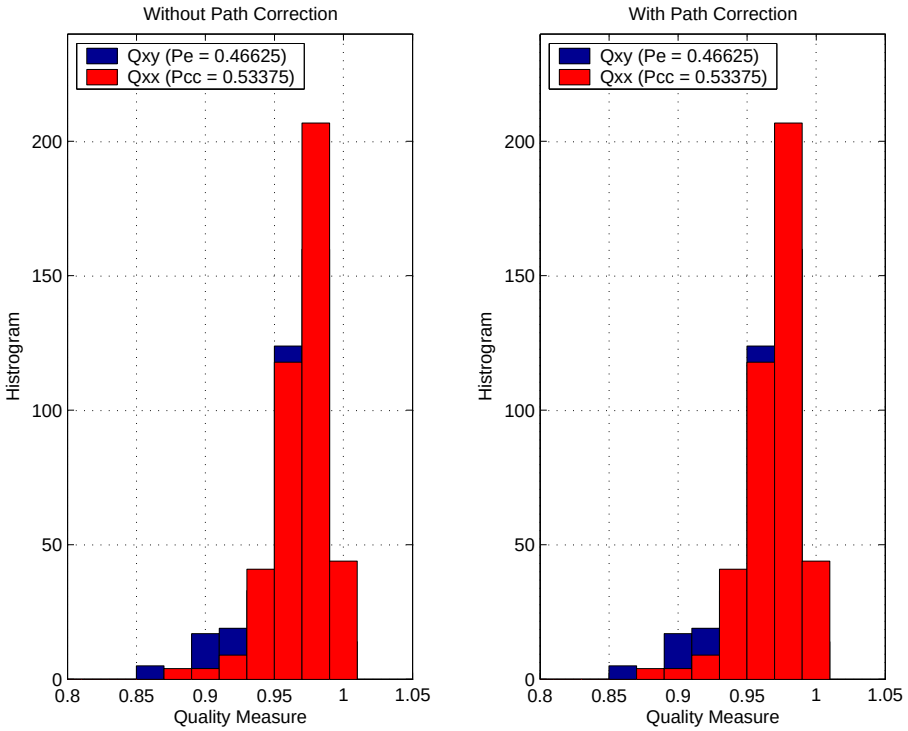


Figure 126: Histogram of quality measures for Experiment 7–Letter data set.

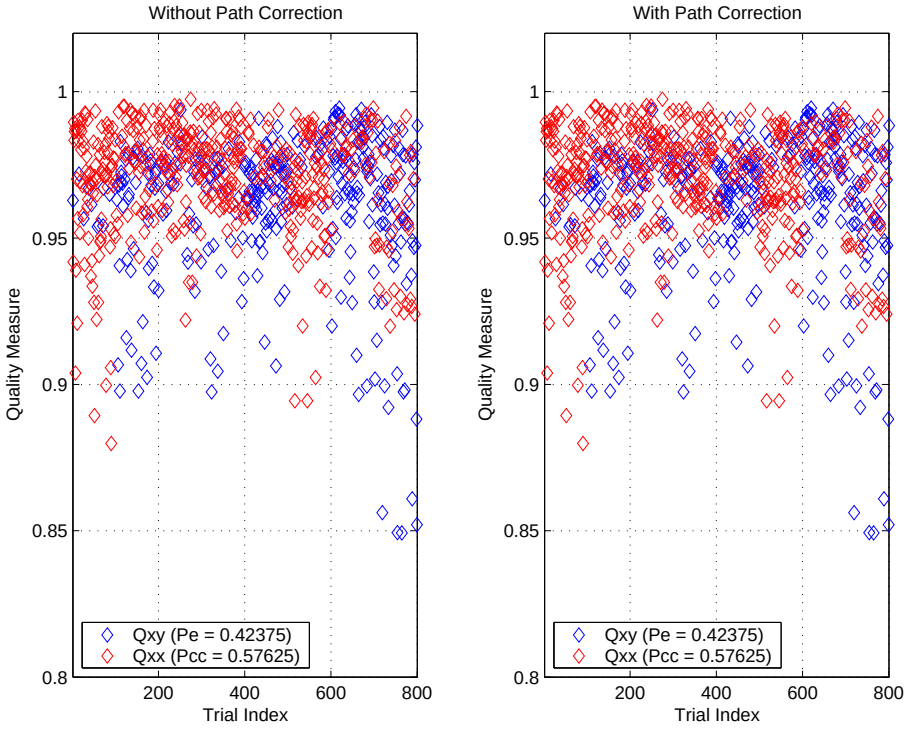


Figure 127: Quality measures for Experiment 8–Letter data set.

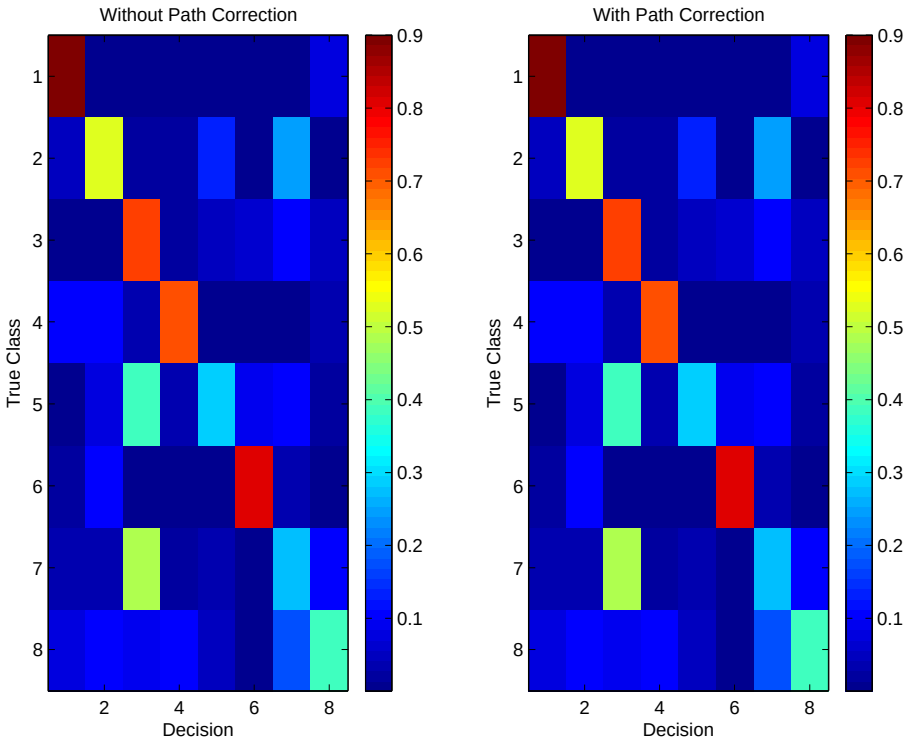


Figure 128: Confusion matrix for Experiment 8–Letter data set.

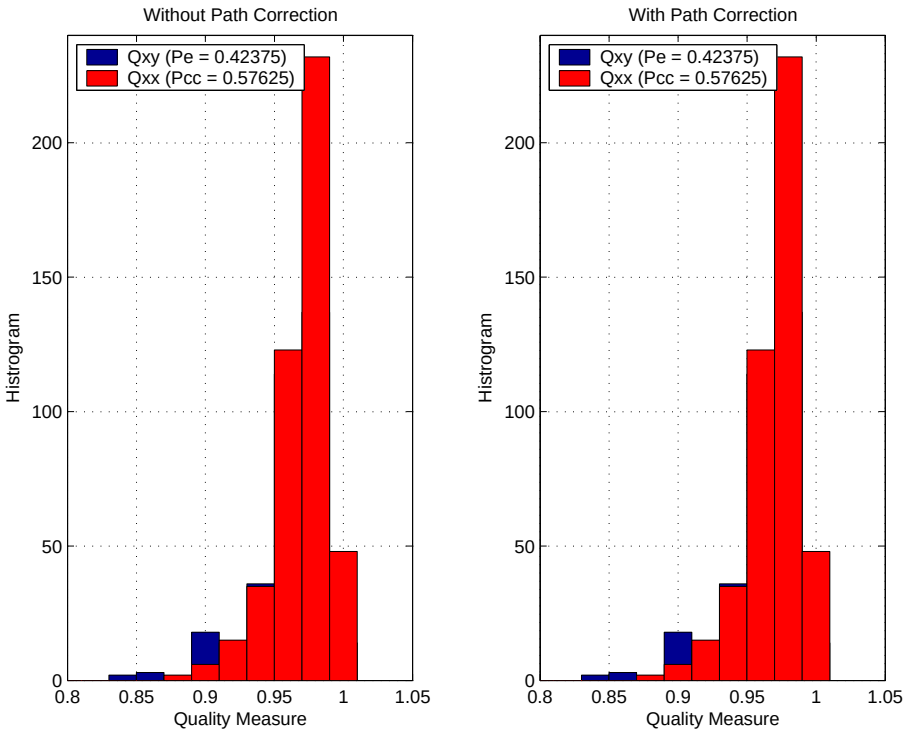


Figure 129: Histogram of quality measures for Experiment 8–Letter data set.

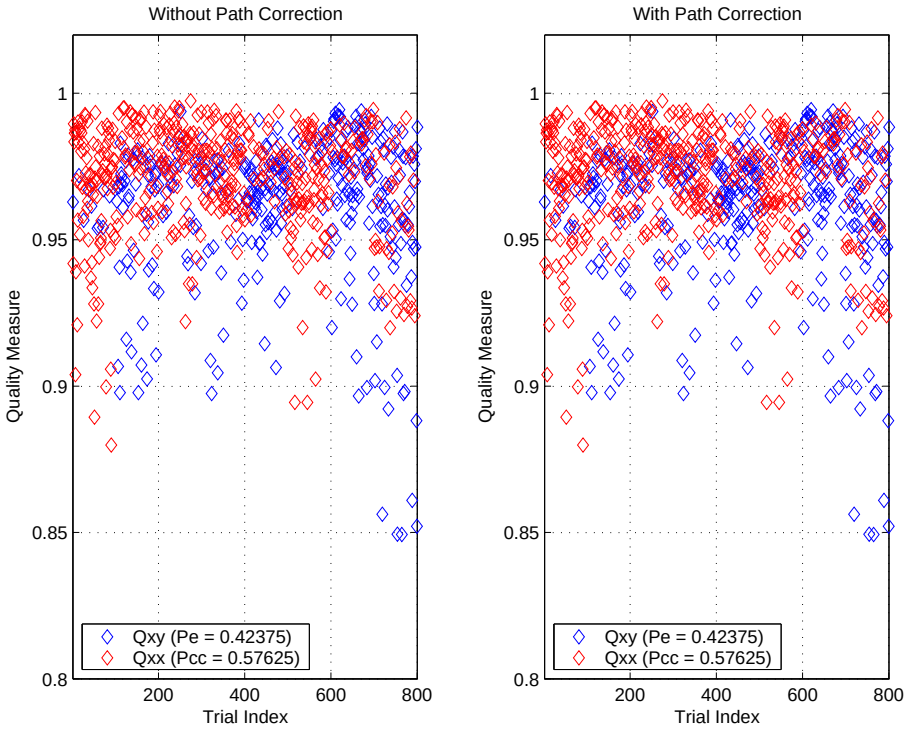


Figure 130: Quality measures for Experiment 9–Letter data set.

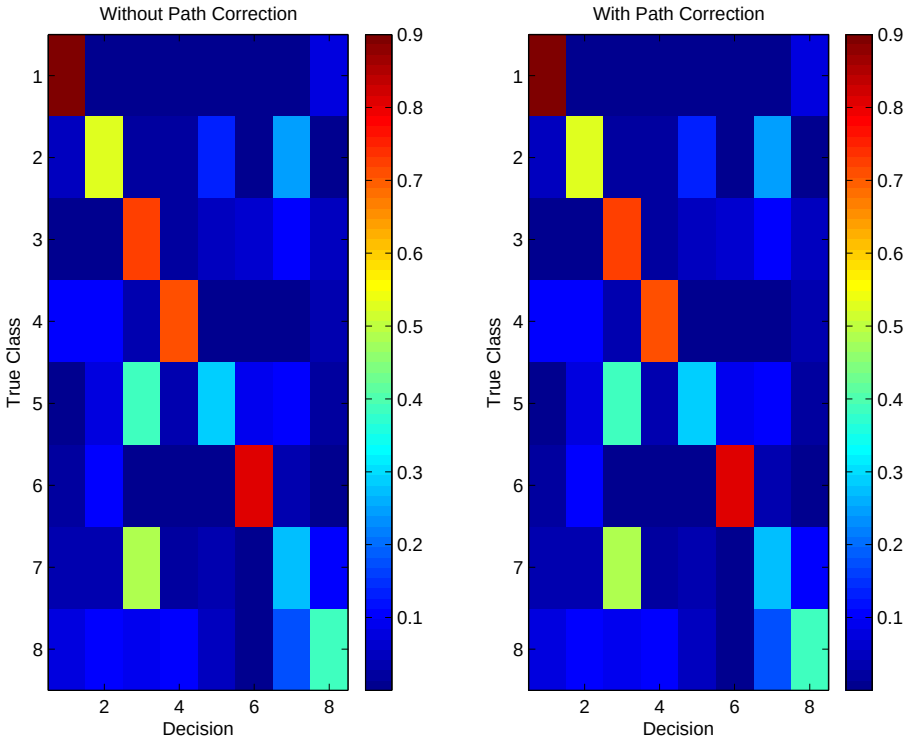


Figure 131: Confusion matrix for Experiment 9–Letter data set.

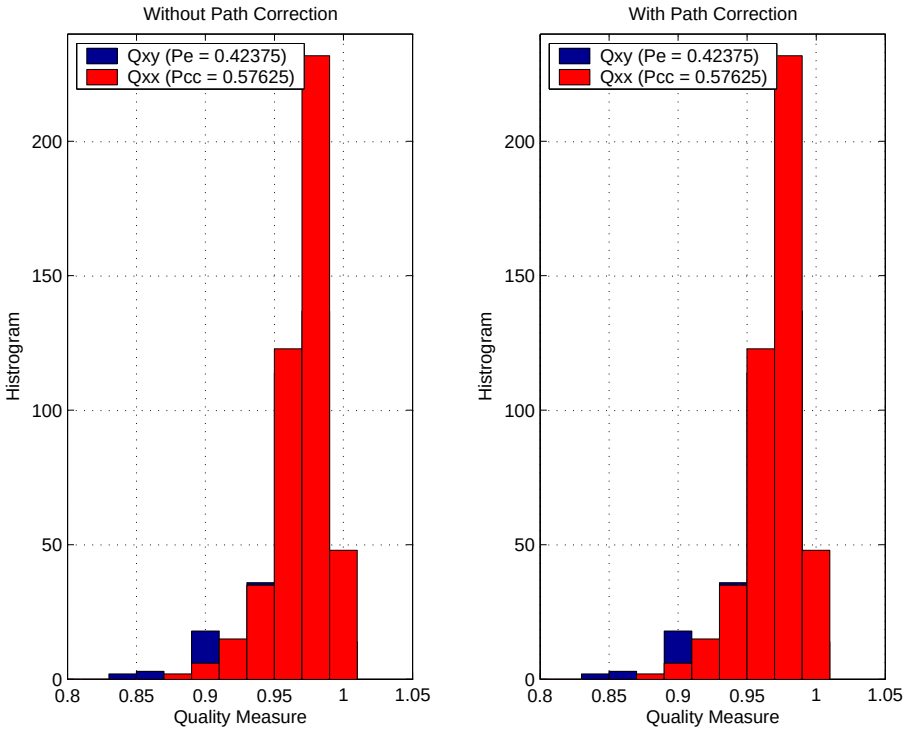


Figure 132: Histogram of quality measures for Experiment 9–Letter data set.

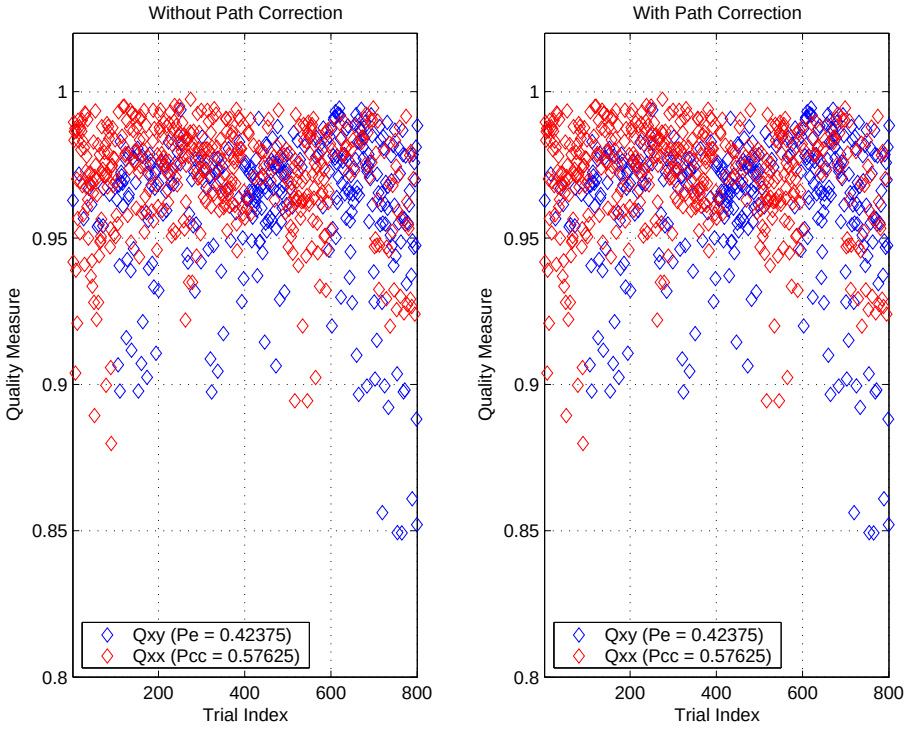


Figure 133: Quality measures for Experiment 10–Letter data set.

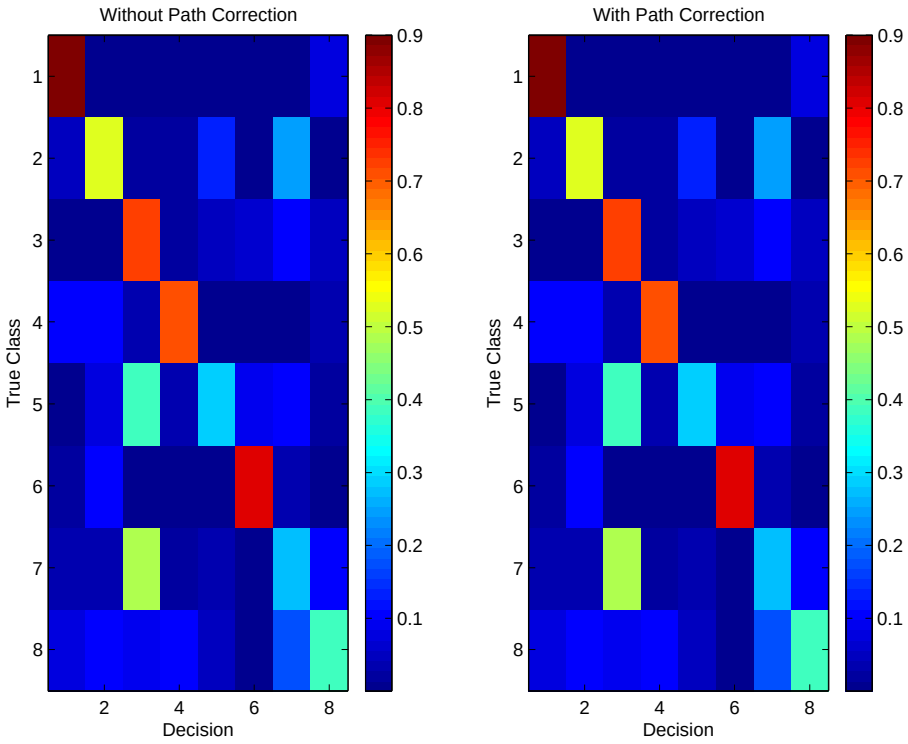


Figure 134: Confusion matrix for Experiment 10–Letter data set.

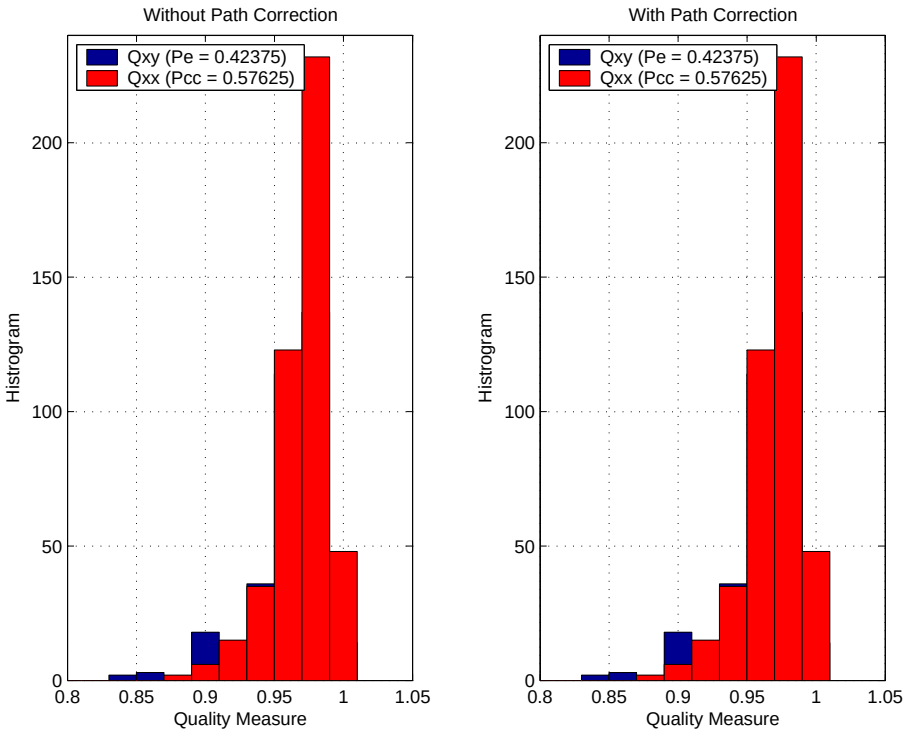


Figure 135: Histogram of quality measures for Experiment 10–Letter data set.

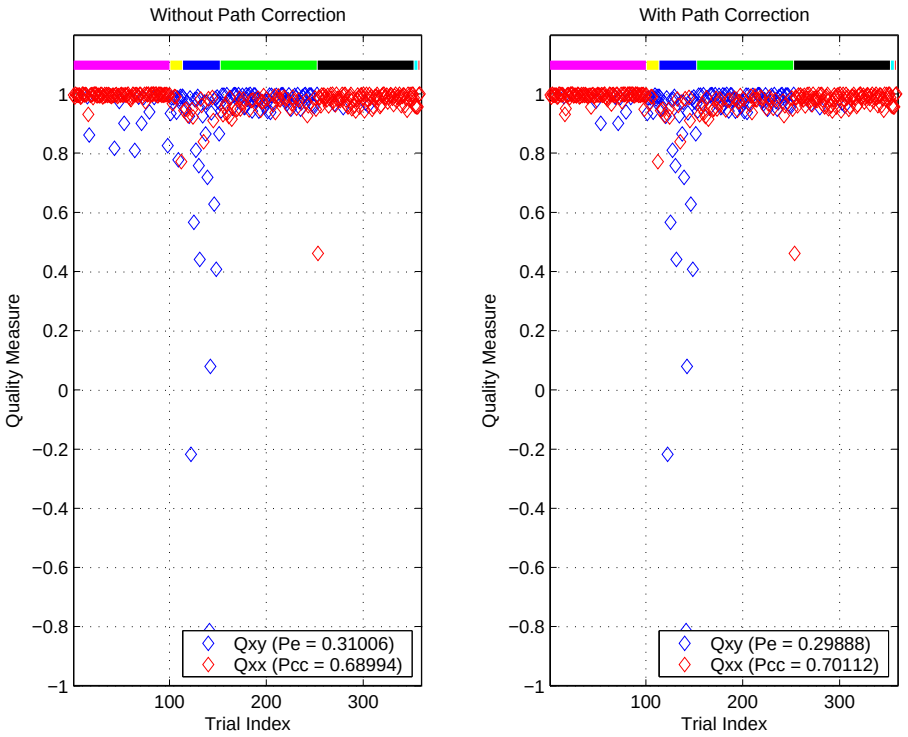


Figure 136: Quality measures for Experiment 11–Shuttle data set.

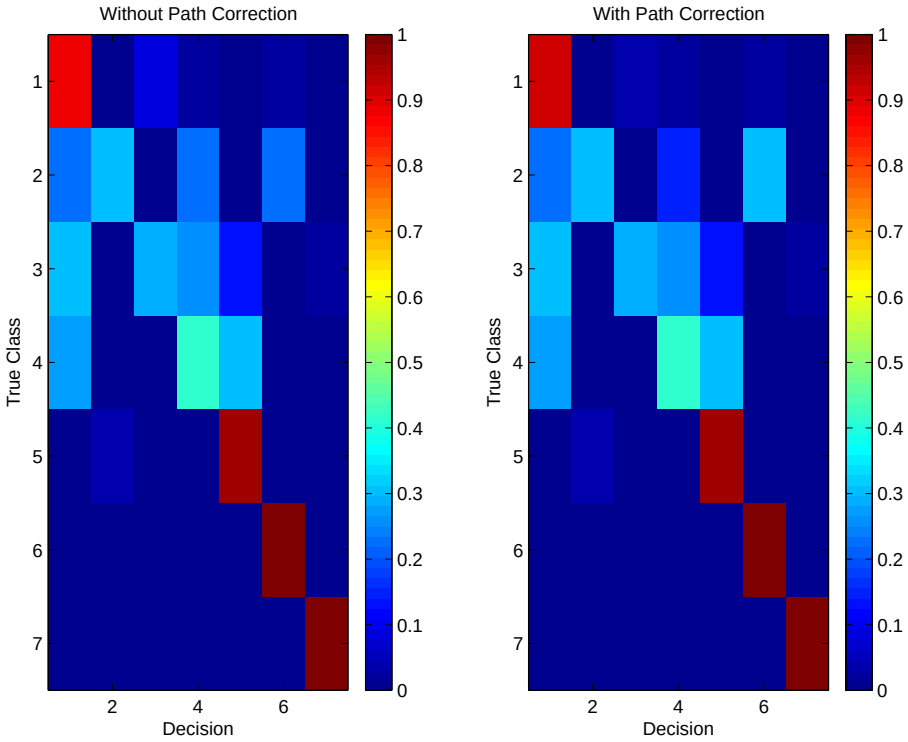


Figure 137: Confusion matrix for Experiment 11–Shuttle data set.

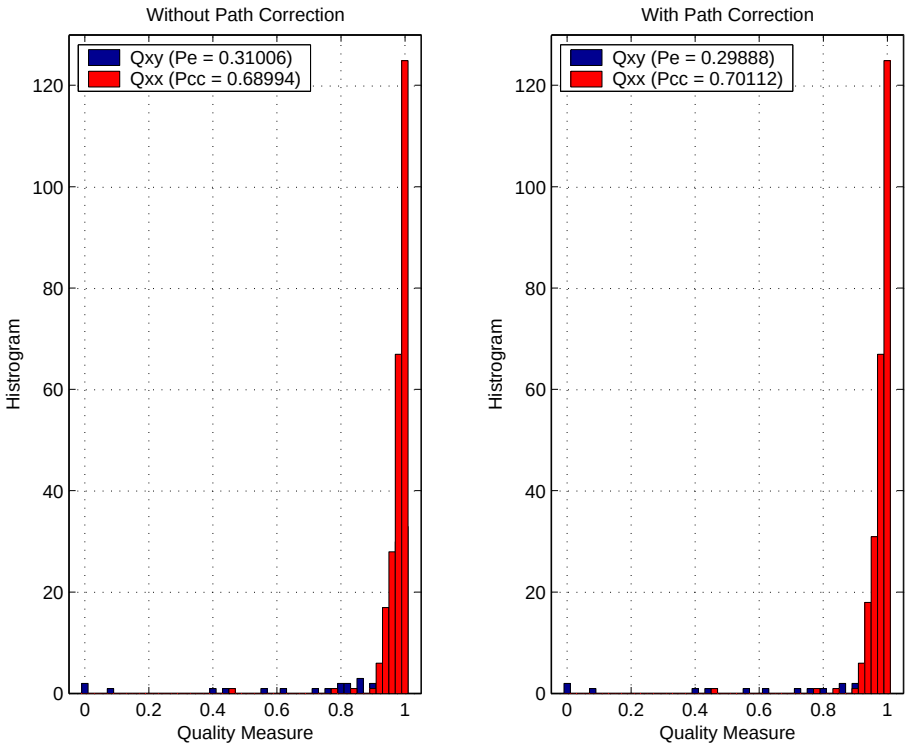


Figure 138: Histogram of quality measures for Experiment 11–Shuttle data set.

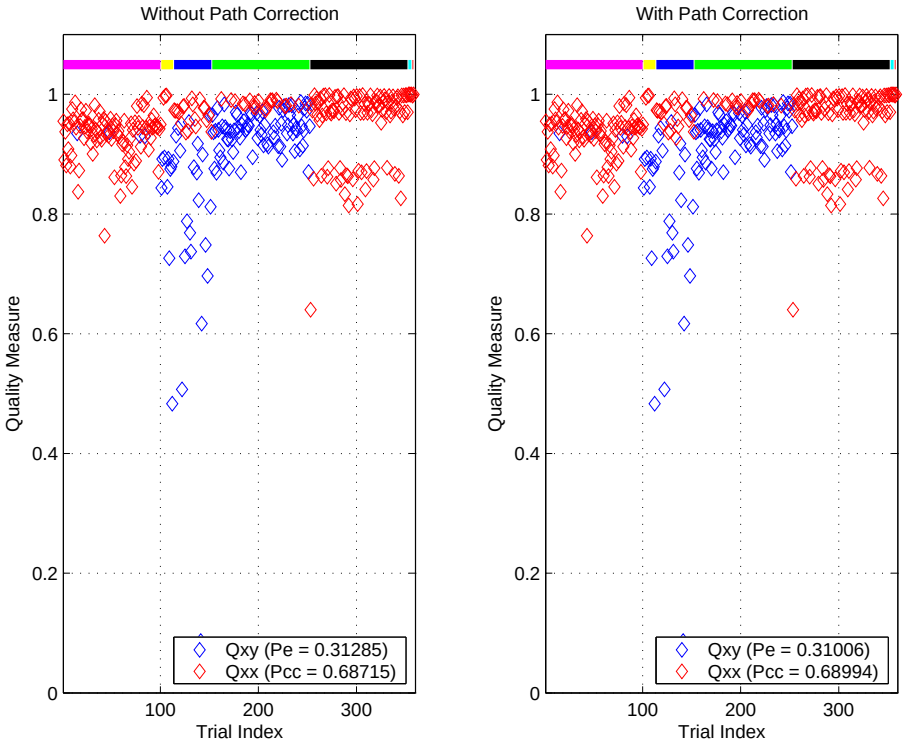


Figure 139: Quality measures for Experiment 12–Shuttle data set.

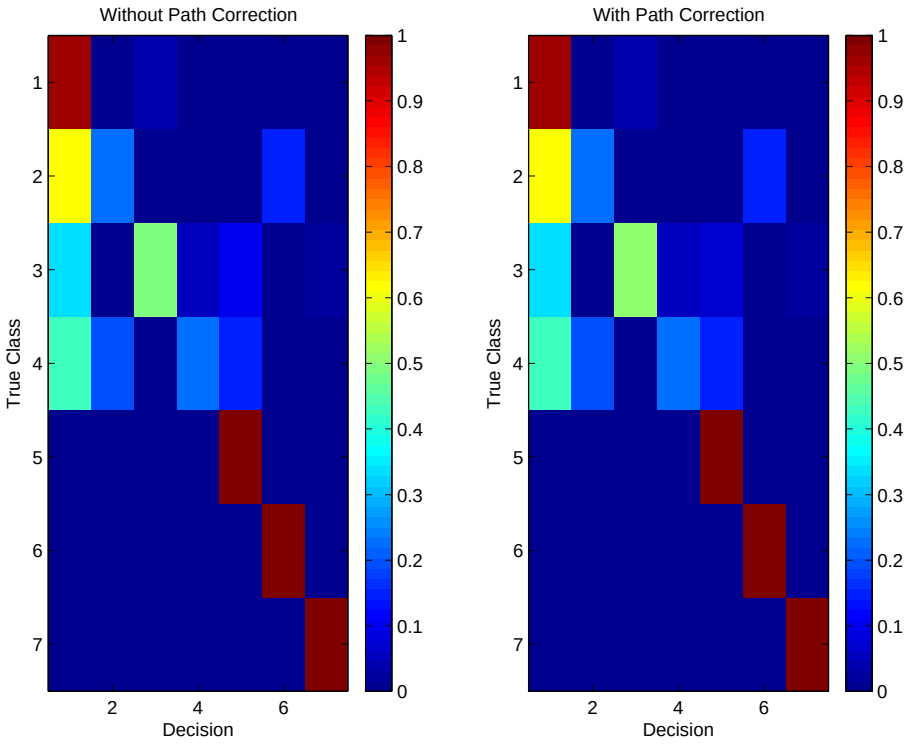


Figure 140: Confusion matrix for Experiment 12–Shuttle data set.

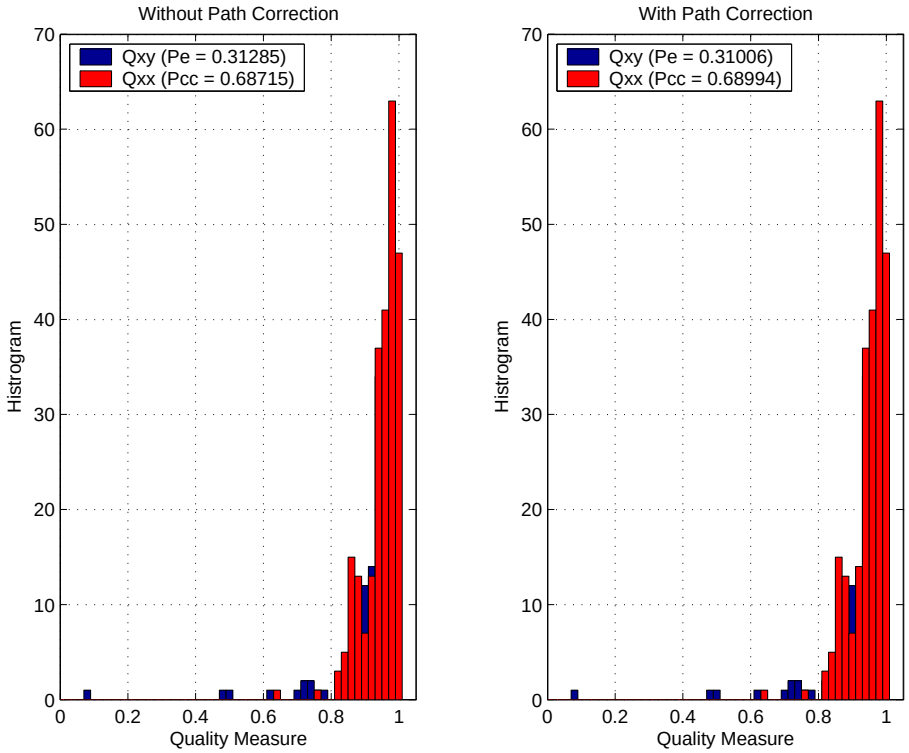


Figure 141: Histogram of quality measures for Experiment 12–Shuttle data set.

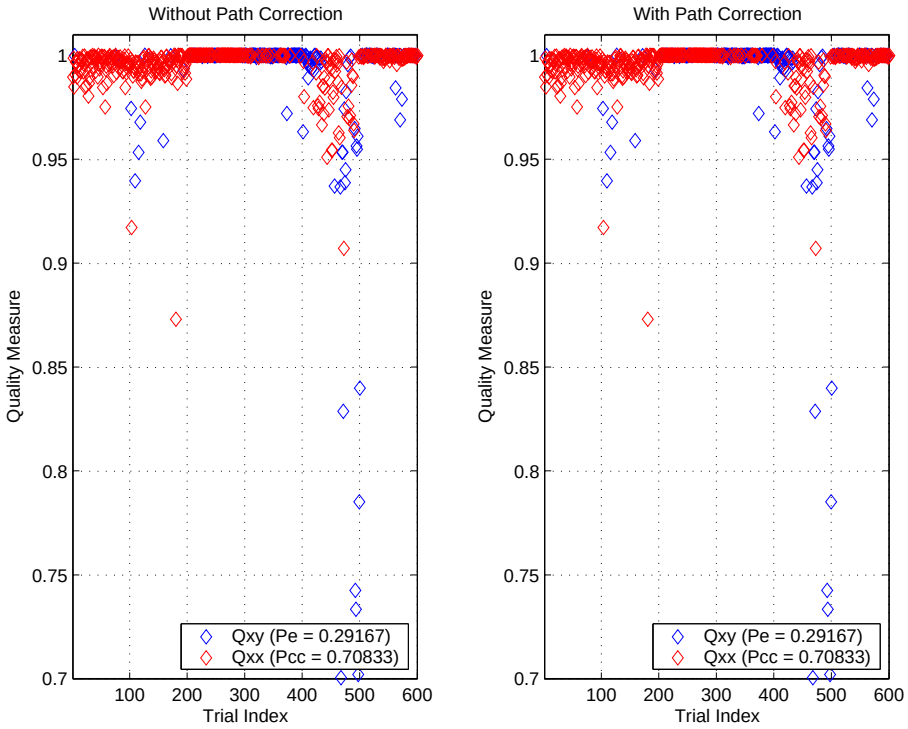


Figure 142: Quality measures for Experiment 13–Satimage data set.

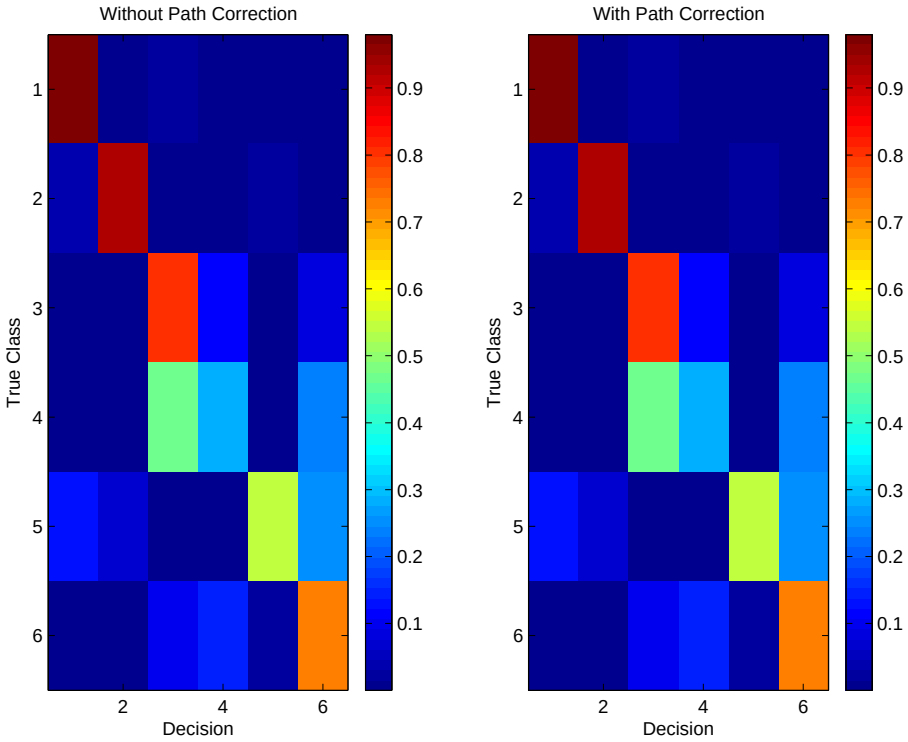


Figure 143: Confusion matrix for Experiment 13–Satimage data set.

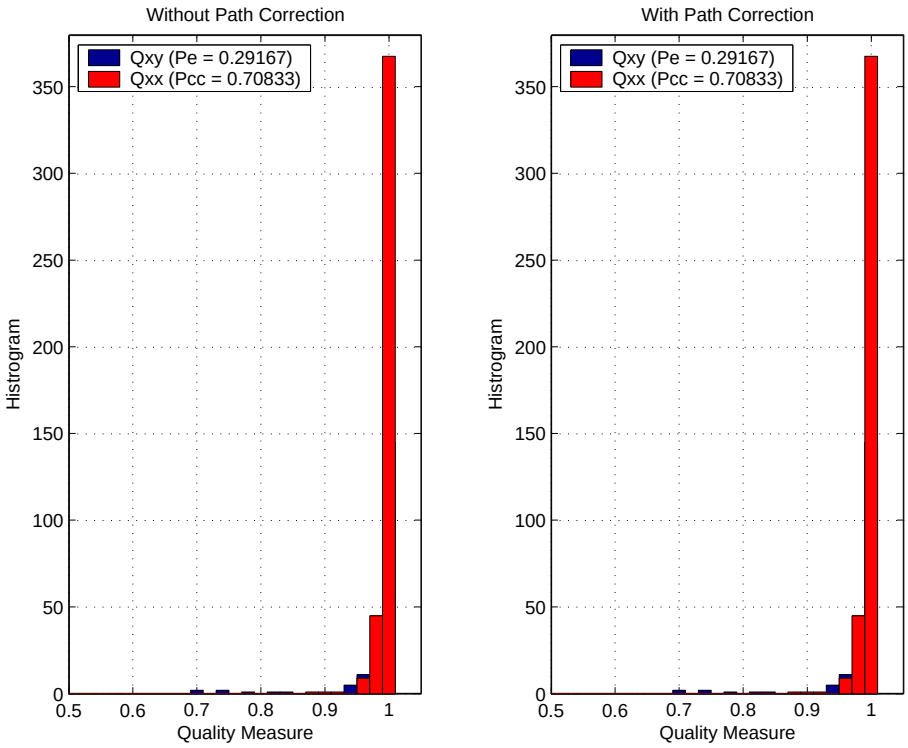


Figure 144: Histogram of quality measures for Experiment 13–Satimage data set.

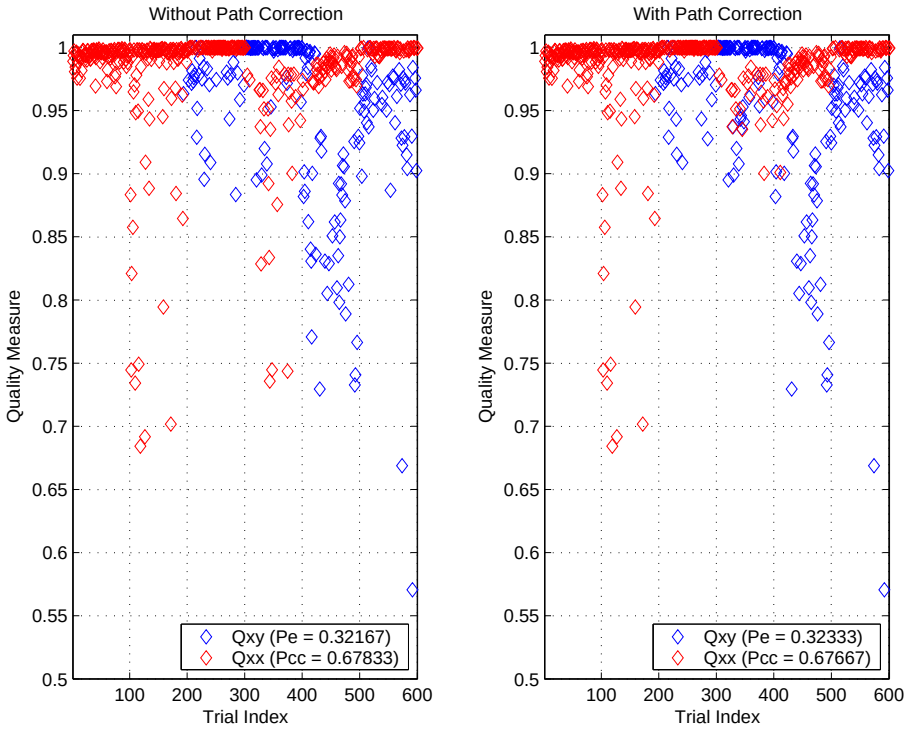


Figure 145: Quality measures for Experiment 14–Satimage data set.

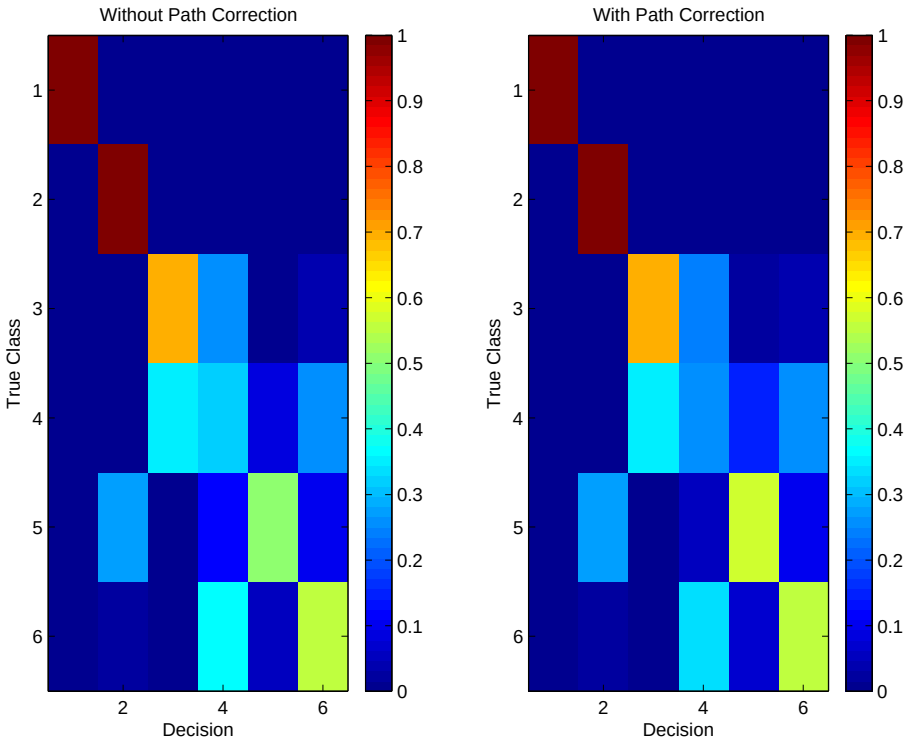


Figure 146: Confusion matrix for Experiment 14–Satimage data set.

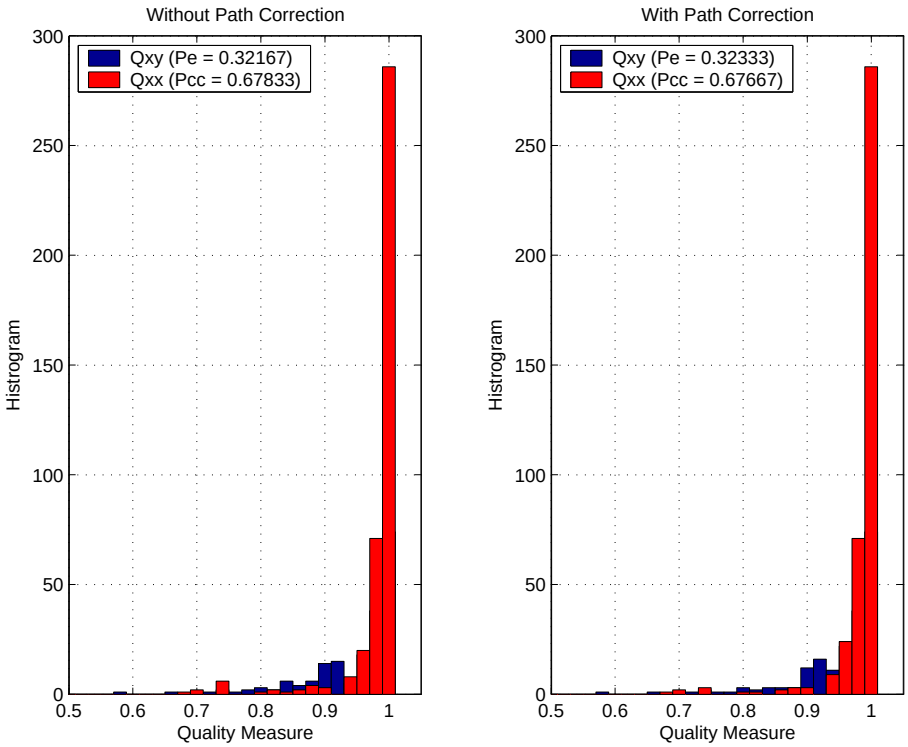


Figure 147: Histogram of quality measures for Experiment 14–Satimage data set.

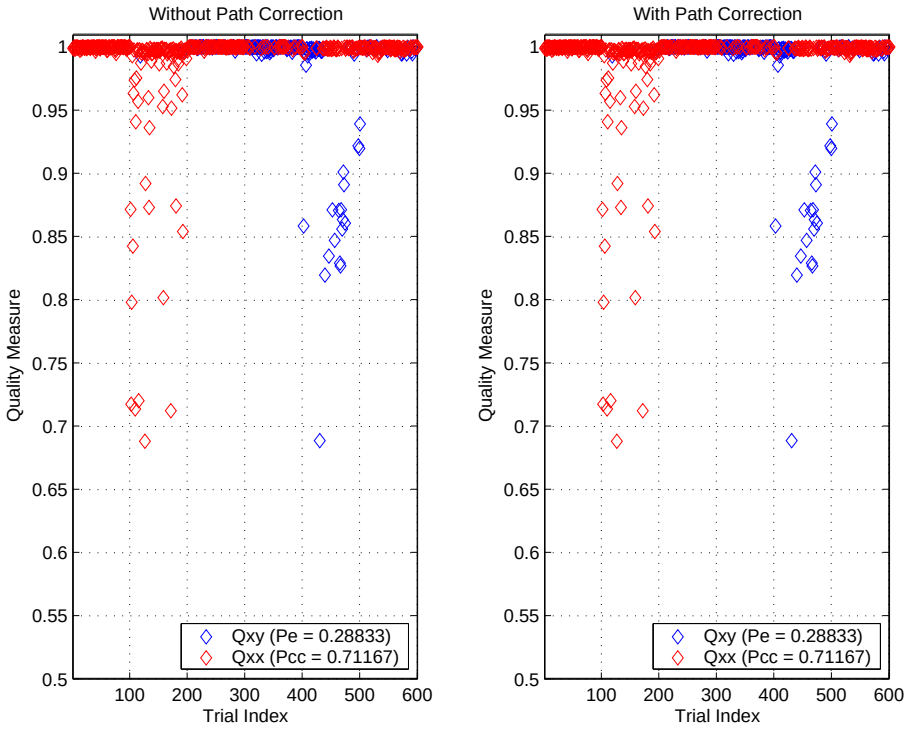


Figure 148: Quality measures for Experiment 15–Satimage data set.

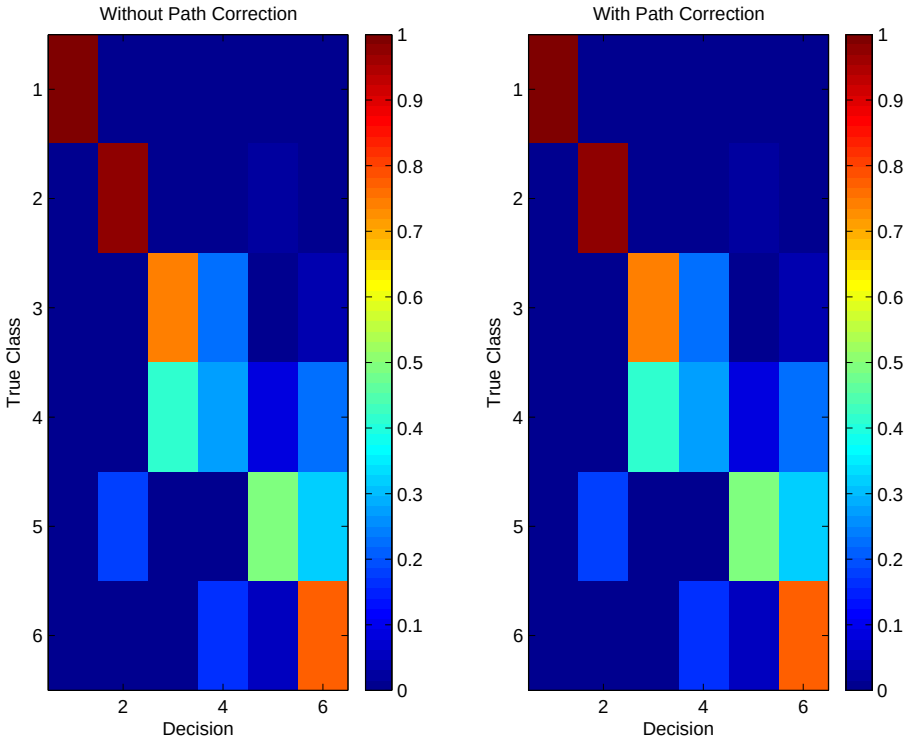


Figure 149: Confusion matrix for Experiment 15–Satimage data set.

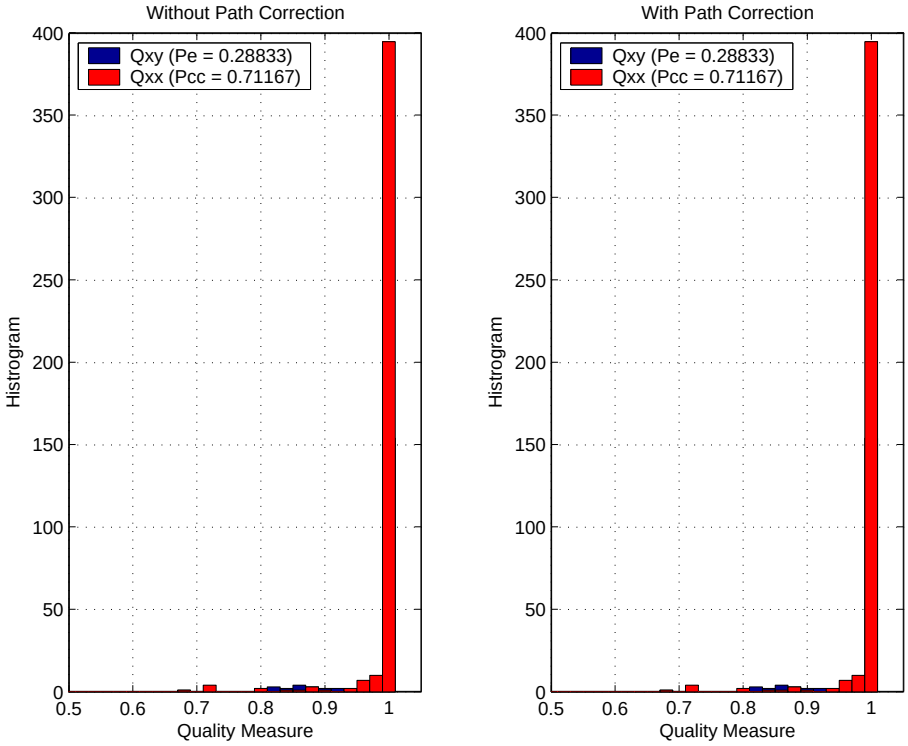


Figure 150: Histogram of quality measures for Experiment 15–Satimage data set.

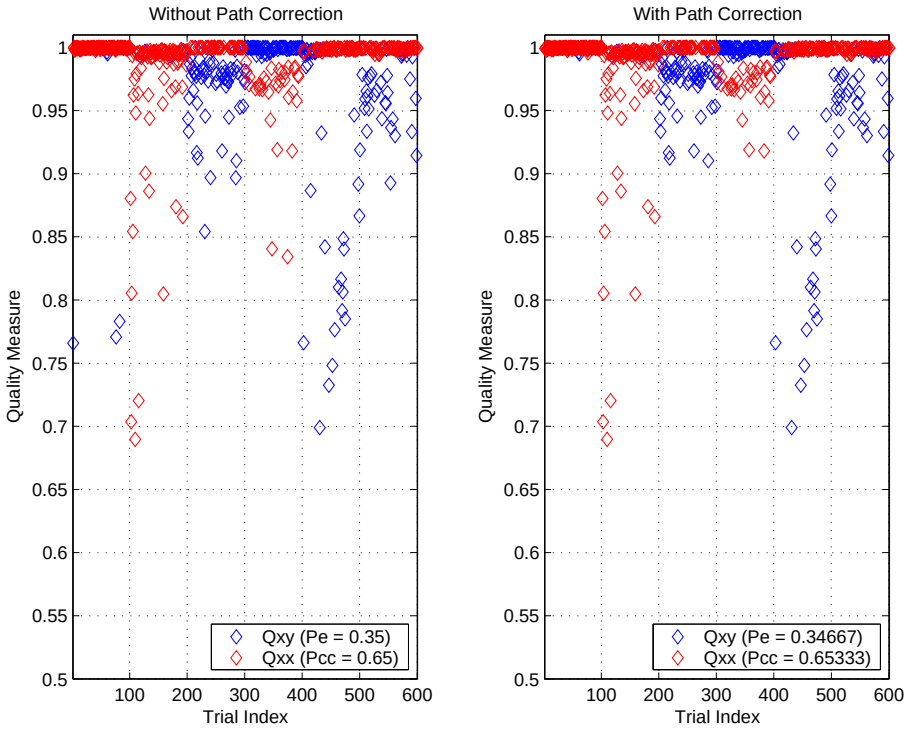


Figure 151: Quality measures for Experiment 16–Satimage data set.

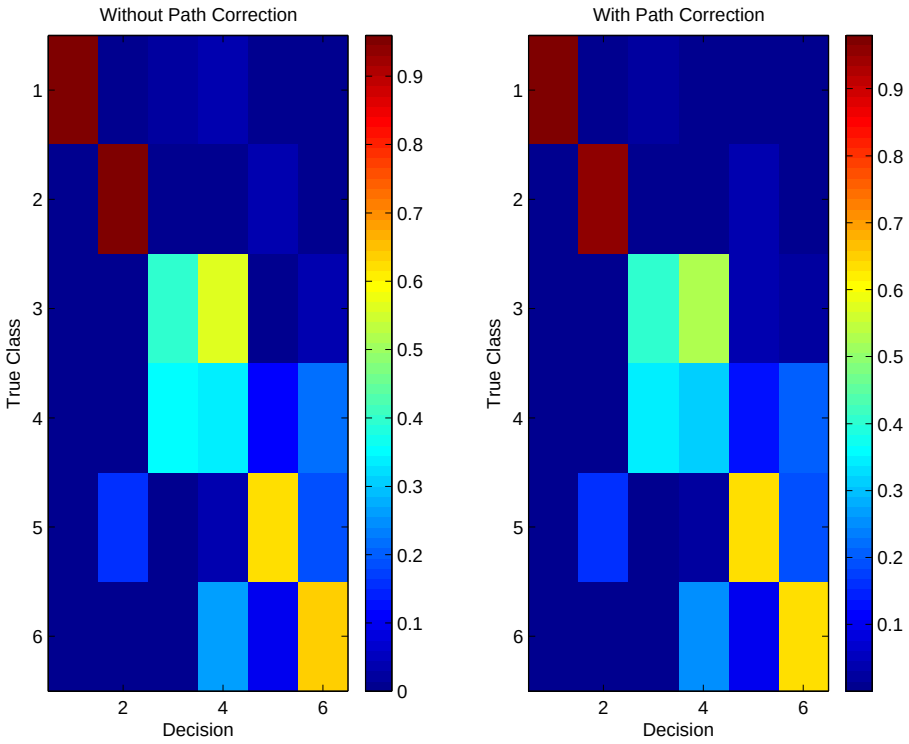


Figure 152: Confusion matrix for Experiment 16–Satimage data set.

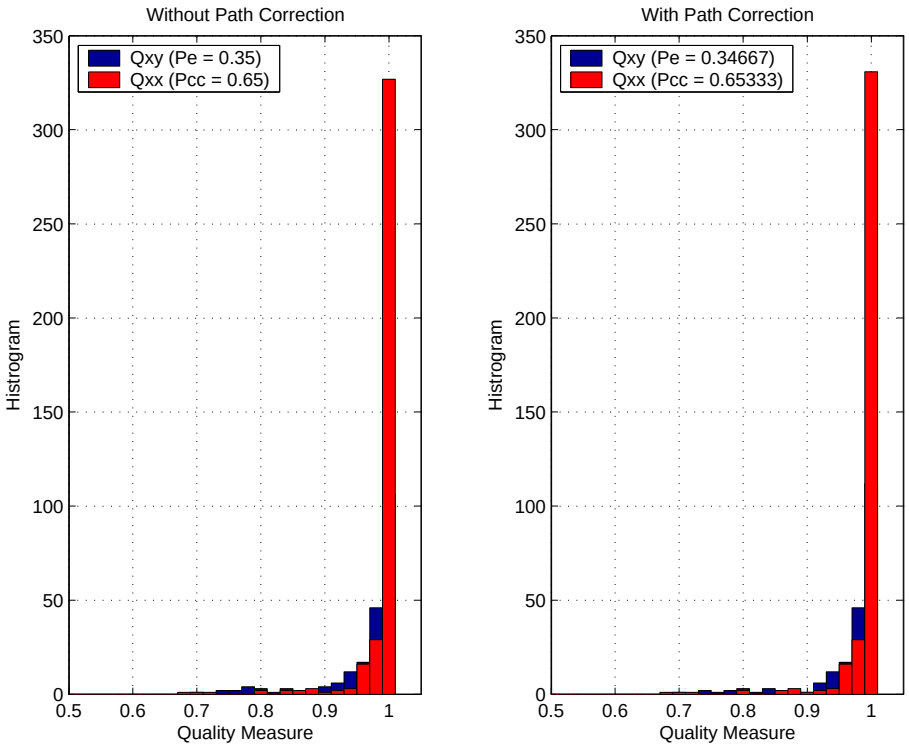


Figure 153: Histogram of quality measures for Experiment 16–Satimage data set.

6 Conclusions

In this final section, we provide a set of concise conclusions that are based on the experimental study of this report.

1. The tree-based classifiers developed in [1], including the *binary-tree classifier* (BTC), *binary hypertree classifier* (BHC), and *binary supertree classifier* (BSC), were studied using synthetic and collected data sets. The basic operation of the classifiers was validated.
2. The algorithm for joint automatic blind specification of the tree's topology, superclass specifications, and feature-vector parameters results in classifiers that are adept at embodying the basic ambiguity structure of the problem at hand. This was demonstrated with synthetic and collected one- and two-dimensional data sets.
3. The basic performance ordering of $\{\text{BTC}\} \leq \text{BHC} \leq \text{BSC}$ was confirmed for the synthetic two-dimensional problem, which contained severe class ambiguities.
4. The basic performance ordering was not confirmed for the synthetic one-dimensional problem (involving automatic recognition of each of sixteen maximal-length shift-register sequences). In this case, an individual BTC could have better performance than the BHC. We conjecture that this contradiction of our mathematical analysis in [1] arises from the breaking of our fundamental assumption that the probability of error at a decision node is a smooth function of the node's ambiguity.
5. The notion of *path correction* was introduced and was shown to dramatically improve performance for problems involving weak class ambiguities and to have little effect (positive or negative) for problems possessing severe class ambiguities.
6. Further work should focus on refinement of the construction of a BHC from constituent BTCs. In particular, the algorithm needs to make better joint use of node ambiguity, node feature-vector strength, and local tree topology in order to improve the selection of nodes to be included as hypertree jump points.

References

[1] C. M. Spooner, "Binary Hypertree Classifiers for ATR: Definitions, Analysis, and Algorithms," MRC Technical Report for the DARPA ISP Program, March 2004.

[2] "A Mathematical Methodology for Managing and Integrating Sensors and Processors in Distributed Systems for Radar and Communications," Mission Research Corporation Proposal for the DARPA ISP Program, October 2001.

[3] C. M. Spooner and G. K. Yeung, "Local Discriminant Bases for TRUMPETS ATR," DARPA TRUMPETS Program, MRC Technical Report, November 2000.

[4] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, Wiley & Sons, New York, 2001.

[5] N. Saito, "Local Feature Extraction and its Applications using a Library of Bases," Ph.D. Dissertation, Yale University, December 1994.

[6] S. Mallat, *A Wavelet Tour of Signal Processing,* Academic Press, San Diego, CA, 1998.

[7] `http://www-stat.stanford.edu/~wavelab.`

[8] S. W. Golomb, *Shift Register Sequences, Revised Edition*, Agean Park Press, Walnut Creek, CA, 1982.

[9] Project StatLog, LIACC, University of Porto, *http://www.liacc.up.pt/ML/statlog/datasets.html.*

[10] The UCI Machine Learning Repository, *http://www.ics.uci.edu/ mlearn/MLRepository.html.*

[11] The Murphy Lab at CMU, *http://murphylab.web.cmu.edu/data/data.html.*